CAMS Project

NOAA/CAMS CFS Implementation Convert Non-CFS Undelivered Orders and Accruals Program Module (FINAL)

Version 2.0

Prepared by Accenture for

U.S. Department Of Commerce
National Oceanic and Atmospheric Administration
Implementation Center
201 Perry Parkway
Gaithersburg, MD 20877

February 01, 2002

Document Control

Change Record

Date	Author	Document Version	Change Reference
01/11/2002	J. Cook	1.0	Draft
01/14/2002	K. Stockmeier	1.1	Draft
01/23/2002	K. Stockmeier	1.2	Draft
01/31/2002	J. Cook	2.0	Final

Reviewer

Name	Position	Date
M. Barron	Manager	01/04/2002
K. Kirwan	Manager	01/08/2002

Approver

Name	Position	Date
W. Holdsworth	Team Lead	01/11/2002

Table of Contents

1 Introduction to Convert Non-CFS Undelivered Orders and Accruals Program Module . . . . Page 1
1.1 Naming Standards . . . . Page 1
1.2 Purpose of Task . . . . Page 1
1.3 Process Flow for the Program Module . . . . Page 1
1.4 Business Rules for the Program Module . . . . Page 4
1.5 Input to the Program Module . . . . Page 5
1.6 Processing of the Program Module . . . . Page 5
1.6.1 Output . . . . Page 5
1.6.2 Error Handling . . . . Page 5

2 Validate Document Data Function . . . . Page 6
2.1 Purpose of Validate Document Data Function . . . . Page 6
2.2 Process Flow of Validate Document Data Function . . . . Page 6
2.3 Business Rules for Validate Document Data Function . . . . Page 7
2.4 Input to Validate Document Data Function . . . . Page 7
2.5 Validate Document Data Function Processing . . . . Page 7
2.5.1 Validate Document Type . . . . Page 7
2.5.1.1 Logic . . . . Page 7
2.5.1.2 Output . . . . Page 8
2.5.2 Validate Item Type . . . . Page 9
2.5.2.1 Logic . . . . Page 9
2.5.2.2 Output . . . . Page 9
2.5.3 Validate Commodity Code . . . . Page 9
2.5.3.1 Logic . . . . Page 9
2.5.3.2 Output . . . . Page 10
2.5.4 Validate Payment Office Code . . . . Page 10
2.5.4.1 Logic . . . . Page 10
2.5.4.2 Output . . . . Page 10

3 Crosswalk Document Data Function . . . . Page 10
3.1 Purpose of Crosswalk Document Data Function . . . . Page 10
3.2 Process Flow of Crosswalk Document Data Function . . . . Page 11
3.3 Input to Crosswalk Document Data Function . . . . Page 13
3.4 Crosswalk Document Data Function Processing . . . . Page 13
3.4.1 Validate Document Will Be Converted . . . . Page 13
3.4.1.1 Logic . . . . Page 13
3.4.2 Crosswalk Document Type . . . . Page 13

3.4.2.1 Logic . . . . Page 13
3.4.2.2 Error Handling . . . . Page 14
3.4.3 Crosswalk Item Type . . . . Page 14
3.4.3.1 Logic . . . . Page 14
3.4.3.2 Error Handling . . . . Page 14
3.4.4 Crosswalk Commodity Code . . . . Page 14
3.4.4.1 Logic . . . . Page 14
3.4.4.2 Error Handling . . . . Page 14
3.4.5 Validate Payment Office Code . . . . Page 14
3.4.5.1 Logic . . . . Page 15
3.4.5.2 Error Handling . . . . Page 15
3.4.6 Validate Vendor Information . . . . Page 15
3.4.6.1 Logic . . . . Page 15
3.4.6.2 Error Handling . . . . Page 15

4 Validate Accounting Data Function . . . . Page 15
4.1 Purpose of Validate Accounting Data Function . . . . Page 16

5 Record Purchase Order Function . . . . Page 16
5.1 Purpose of Record Purchase Order Function . . . . Page 16
5.2 Process Flow of Record Purchase Order Function . . . . Page 16
5.3 Record Purchase Order Function Processing . . . . Page 18
5.3.1 Initialize Change Order To ‘0' . . . . Page 18
5.3.1.1 Logic . . . . Page 18
5.3.1.2 Output . . . . Page 18
5.3.2 Record Purchase Order Control Block . . . . Page 18
5.3.2.1 Logic . . . . Page 19
5.3.2.2 Output . . . . Page 29
5.3.2.3 Error Handling . . . . Page 29
5.3.3 Record Purchase Order Item Block . . . . Page 30
5.3.3.1 Logic . . . . Page 30
5.3.3.2 Output . . . . Page 34
5.3.3.3 Error Handling . . . . Page 35
5.3.4 Record Purchase Order Account Block . . . . Page 35
5.3.4.1 Logic . . . . Page 35
5.3.4.2 Output . . . . Page 43
5.3.4.3 Error Handling . . . . Page 43
5.3.5 Record Purchase Order General Ledger . . . . Page 44
5.3.5.1 Logic . . . . Page 44
5.3.5.2 Error Handling . . . . Page 49
5.3.6 Is There Another Line Item For This Fiscal Year . . . . Page 49

5.3.6.1 Logic . . . . Page 49
5.3.7 Record Purchase Order Approval . . . . Page 50
5.3.7.1 Logic . . . . Page 50
5.3.7.2 Error Handling . . . . Page 50
5.3.8 Is There Another Line Item . . . . Page 51
5.3.8.1 Logic . . . . Page 51
5.3.9 Increment Change Order by 1 . . . . Page 51
5.3.9.1 Logic . . . . Page 51
5.3.10 Determine If A Matching EA Will Be Created . . . . Page 51
5.3.10.1 Logic . . . . Page 51

6 Record Estimated Accrual Function . . . . Page 52
6.1 Purpose of Record Estimated Accrual Function . . . . Page 52
6.2 Process Flow of Record Estimated Accrual Function . . . . Page 52
6.3 Record Estimated Accrual Function Processing . . . . Page 54
6.3.1 Record Estimated Accrual Control Block Function . . . . Page 54
6.3.1.1 Logic . . . . Page 54
6.3.1.2 Output . . . . Page 58
6.3.1.3 Error Handling . . . . Page 58
6.3.2 Record Estimated Accrual Item Block . . . . Page 58
6.3.2.1 Logic . . . . Page 58
6.3.2.2 Output . . . . Page 61
6.3.2.3 Error Handling . . . . Page 61
6.3.3 Record Estimated Accrual Account Block . . . . Page 62
6.3.3.1 Logic . . . . Page 62
6.3.3.2 Output . . . . Page 66
6.3.3.3 Error Handling . . . . Page 66
6.3.4 Record General Ledger . . . . Page 66
6.3.4.1 Logic . . . . Page 66
6.3.4.2 Output . . . . Page 72
6.3.4.3 Error Handling . . . . Page 72
6.3.5 Is There Another Line Item? . . . . Page 72
6.3.5.1 Logic . . . . Page 73

7 Record Invoice Shell Function . . . . Page 73
7.1 Purpose of Record Invoice Shell Function . . . . Page 73
7.2 Process Flow of Invoice Shell Order Function . . . . Page 74
7.3 Business Rules of Record Invoice Shell Function . . . . Page 74
7.4 Input to Record Invoice Shell Function . . . . Page 75
7.5 Record Invoice Shell Function Processing . . . . Page 75
7.5.1 Record Invoice Shell Control Block . . . . Page 75

7.5.1.1 Logic . . . . Page 75
7.5.1.2 Error Handling . . . . Page 79
7.5.2 Record Invoice Shell Detail Block . . . . Page 79
7.5.2.1 Logic . . . . Page 79
7.5.2.2 Error Handling . . . . Page 84
7.5.3 Is There Another Unprocessed Estimated Accrual Line Item . . . . Page 85
7.5.3.1 Logic . . . . Page 85

8 Core Financial System (CFS) Set-Up . . . . Page 85
8.1 Document Type . . . . Page 85
8.2 Item Type . . . . Page 85
8.3 Prompt Pay Type . . . . Page 85
8.4 Commodity Code . . . . Page 86
8.5 Invoice Type . . . . Page 86
8.6 Employee Information . . . . Page 86
8.7 Document Matching . . . . Page 86
8.8 Document Tolerance . . . . Page 86

9 Overall Risks . . . . Page 86

10 Overall Issues . . . . Page 87

11 Assumptions . . . . Page 90

12 References . . . . Page 90

Appendix A . . . . Page 91

Appendix B . . . . Page 95

Appendix C . . . . Page 100

Appendix D . . . . Page 103

Appendix E . . . . Page 104

1 Introduction to Convert Non-CFS Undelivered Orders and Accruals Program Module

The purpose of this document is to provide a detailed account of how Non-CFS documents will be converted from the National Oceanic and Atmospheric Administration's (NOAA's) financial management system (FIMA) to the Commerce Administrative Management System Core Financial System (CAMS CFS).

1.1 Naming Standards

The Convert Non-CFS Undelivered Order and Accruals Program Module adheres to the latest version of the CAMS Support Center (CSC) Programming Standards and Guidelines.

Throughout this document suggested names of objects are consistently formatted in order to make it easier to distinguish what type of object is being referred to in the design. The following table is a list of each object in this document and its corresponding format.

Object	Description
SCREENS	Named in all caps and italics
Field Names	Named in italics with the first letter of each word capitalized
TABLES	Named in all caps and bold
Column Names	Named in bold with the first letter of each word capitalized
FILES	Named in all caps and underlined
variables	Named in all lowercase and underlined

1.2 Purpose of Task

The Convert Non-CFS Undelivered Order and Accruals Program Module is one of several documents that describe the detail-level design of the automated accounts payable open document conversion.

1.3 Process Flow for the Program Module

This section describes a high level overview of the major steps needed to transfer Non-CFS documents from **FIMA ACTDOC** to CFS. The data conversion is comprised of the steps illustrated in Figure 1.

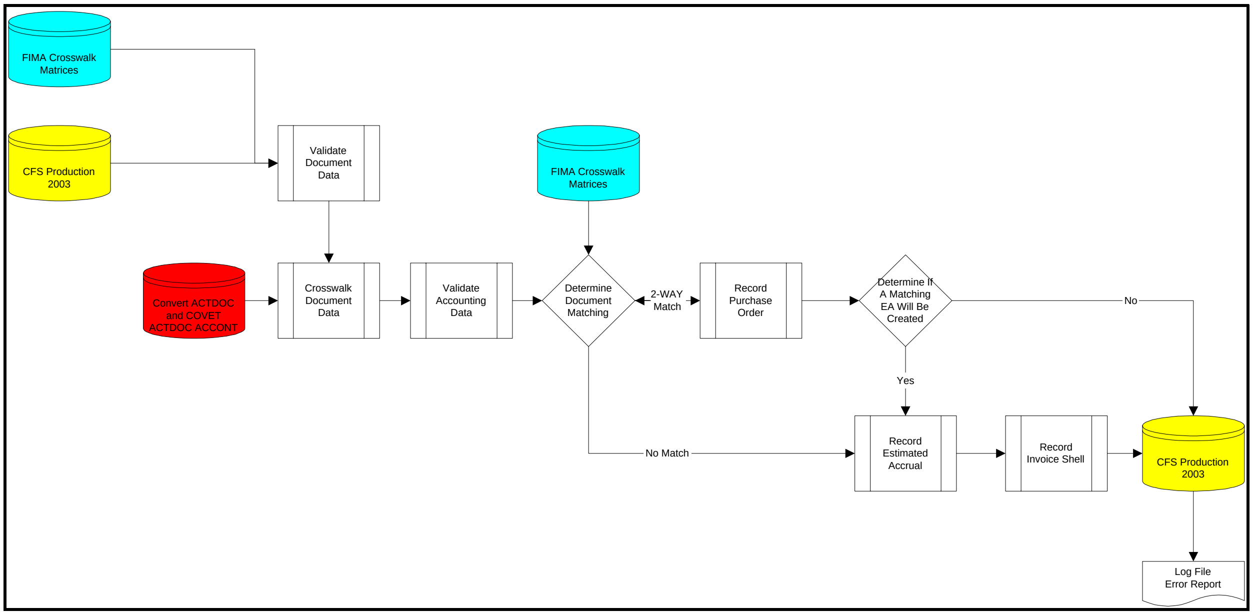

Figure 1. High Level Process Flow

Validate Document Data. The Program Module begins by validating the document data needed for conversion. This step is only executed once each time the Program Module is initiated. Data that is validated is appropriately marked.

Crosswalk Document Data. Once the document data has been validated the Program Module selects the first **FIMA Document Number** (**Document_no**) eligible for conversion and all the associated data elements from **CONVERT ACTDOC** (**CONV_ACTDOC** and **CONV_ACTDOC_ACCOUNT**). Criteria for document eligibility is presented in the Crosswalk Document Data Function. The **Document Sequence** (**Document_sequence**) numbers for each **FIMA Document Number** are ordered by **Fiscal Year** (**Fiscal_year**) to simplify insertion into CFS tables. This step is executed for each **FIMA Document Number** eligible for conversion.

Validate Accounting Data. The Program Module must validate the accounting data for the selected **FIMA Document Number** and all associated data elements. Any accounting data that fails validation will cause the Program Module to abort converting the selected **FIMA Document Number**, and select the next **FIMA Document Number** eligible for conversion. This step is executed for each **FIMA Document Number** eligible for conversion.

Determine Document Matching. Once all the data elements have been validated for the selected **FIMA Document Number**, the Program Module determines the document matching of the selected **FIMA Document Number**. The document matching identifies whether the Program Module creates a Purchase Order or an Estimated Accrual. Appendix A displays the document matching matrix that will drive this step. This step is executed for each **FIMA Document Number** eligible for conversion.

Record Purchase Order. A **FIMA Document Number** identified as a 2-Way match document requires the Program Module to create a Purchase Order. The Program Module creates a Purchase Order and CFS General Ledger entries as outlined in the Record Purchase Order Function. This step is executed for each **FIMA Document Number** eligible for conversion and identified as a 2-Way match document.

Determine If A Matching EA Will Be Created. Upon completion on the Record Purchase Order Function, the Program Module determines if an Estimated Accrual is created. The determination is based on the criteria presented in the Determine If A Matching EA Will Be Created Sub-Function of the Record Purchase Order Function. If the Program Module determines that an Estimated Accrual must be created for the selected **FIMA Document Number**, the Record Estimated Accrual Function is executed. Otherwise, the appropriate flags are updated and the Program Module chooses the next **FIMA Document Number** eligible for conversion and all associated data elements. This step is executed for each **FIMA Document Number** eligible for conversion and identified as a 2-Way match document.

Record Estimated Accrual. When the Program Module determines that a selected document is identified as a No-Match document or the Program Module determines that an Estimated Accrual must be created, the Record Estimated Accrual Function is executed. The Program Module creates an Estimated Accrual and CFS General Ledger entries as outlined in the Record Estimated Accrual Function. This step is executed for each **FIMA Document Number** eligible for conversion and identified as a No-Match document or a 2-Way match document that requires an Estimated Accrual.

Record Invoice Shell. Any time the Record Estimated Accrual Function is completed, the Program Module executes the Record Invoice Shell Function. The Record Invoice Shell Function details how the Function is executed. This step is executed for each **FIMA Document Number** eligible for conversion and invokes the Record Estimated Accrual Function.

Upon completion of data conversion, the Program Module will produce a log file and error report. The log file displays statistics of the conversion, while the error report identifies which documents were not converted and provides a detailed explanation.

1.4 Business Rules for the Program Module

The following business rules pertain to this Program Module:

- CFS requires that distinct Purchase Order accounting lines and fiscal year combinations within a given **CFS Document Number** be differentiated by the **Change Order Number** (**Change_order_no**).

- CFS requires the Estimated Accrual to reference the latest Purchase Order **Change Order Number**.

- CFS requires a valid vendor on any approved document.

- CFS requires a valid employee on any approved document.

- A 2-Way match document must be associated with a Purchase Order.

- All documents will be converted as either 2-Way match or No-Match documents. A limited number of **FIMA Document Types** (**Document_type**) are currently processed as 3-Way match documents. 3-Way match documents require a receiving ticket in addition to an obligation. The absence of a valid **Received Date** (**Received_date**) in FIMA prevents the CAMS Design and Conversion Team from creating 3-Way match documents. FIMA Document Types currently processed as 3-Way match documents will be converted as 2-Way match documents during the FIMA-CFS conversion.

- Upon the approval of an Estimated Accrual, an unapproved Invoice Shell is created on the *Vendor Invoice Transaction Screen (PM003).* The Invoice Shell remains unapproved until a user manually approves the document in CFS.

- All approved Purchase Orders and Estimated Accruals produce updates to the CFS General Ledger.

1.5 Input to the Program Module

CONVERT ACTDOC and **CONVERT ACTDOC ACCOUNT** provide the document and accounting information to the Program Module. Documents which have reconciled with CFS or were previously successfully converted will not be considered. In addition no Grants documents will be processed via the Program Module. The following FIMA document data crosswalks and **CONVERT ACTDOC** tables provide the input needed to execute the Program Module.

- FIMA Document Matching Matrix. See Appendix A.
- FIMA Item Type Matrix. See Appendix B.
- FIMA Commodity Code Matrix. See Appendix C.
- FIMA Payment Office Code Matrix. See Appendix D.
- **CONVERT ACTDOC**
- **CONVERT ACTDOC ACCOUNT**

1.6 Processing of the Program Module

1.6.1 Output

For documents that are converted, the Convert Non-CFS Undelivered Order and Accruals Program Module, the Program Module will:

- Update the **Conversion Status Flag** (**Conv_status_flag**) on **CONVERT ACTDOC** with a value of 'Y'.
- Update the **Conversion Date** (**Conv_date**) on **CONVERT ACTDOC** with the current date.

For documents that were not converted, the Convert Non-CFS Undelivered Order and Accruals Program Module, the Program Module will:

- Update the **Conversion Status Flag** (**Conv_status_flag**) on **CONVERT ACTDOC** with a value of 'N'.

1.6.2 Error Handling

If an error occurs at any point after the data validation for a selected **FIMA Document Number**, the Program Module aborts conversion for the selected **FIMA Document Number**. Any data that has already been inserted to any table that references the aborted **FIMA Document Number**, is removed prior to selecting the next FIMA Document Number to be converted. Converted data is only kept in a CFS table if the Program Module is able to successfully convert the entire document.

2 Validate Document Data Function

2.1 Purpose of Validate Document Data Function

The purpose of the Validate Document Data Function is to ensure that the data displayed in Appendices A thru D is present and active in the appropriate table in CFS. This Function is executed prior to selecting a FIMA **Document Number** for conversion and is only performed once during the Open Document Conversion Undelivered Order and Unpaid Accrued Expenditure Program Module.

2.2 Process Flow of Validate Document Data Function

This section describes an overview of the major steps needed to perform the Validate Document Data Function. The Function is comprised of the steps illustrated in Figure 2.

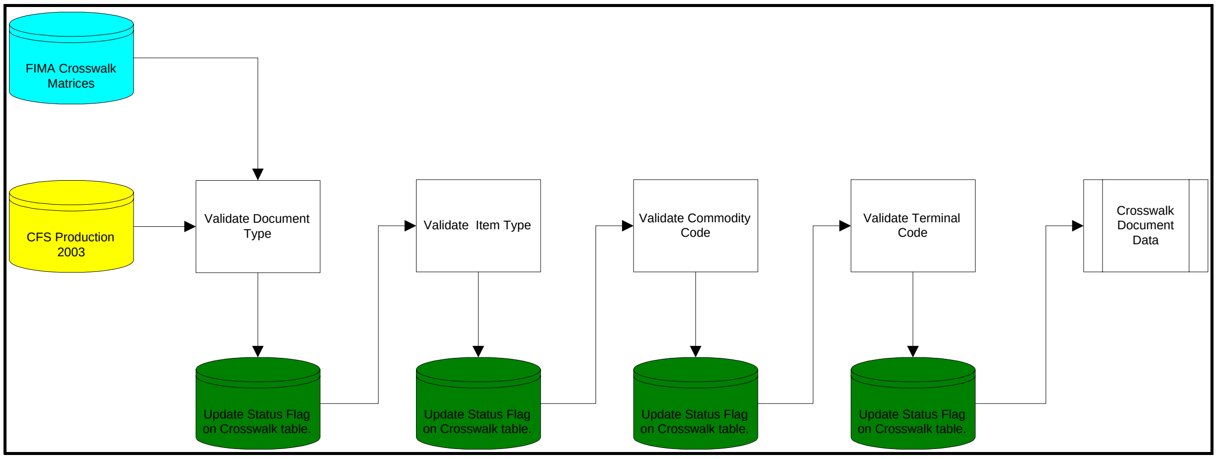
FIMA Crosswalk Matrices
CFS Production 2003
Validate Document Type
Validate Item Type
Validate Commodity Code
Validate Terminal Code
Crosswalk Document Data
Update Status Flag on Crosswalk table.
Update Status Flag on Crosswalk table.
Update Status Flag on Crosswalk table.
Update Status Flag on Crosswalk table.

Figure 2. Validate Document Data Process Flow

Validate Document Type. The Program Module validates that the CFS Document Type values displayed in Appendix A are present and active in CFS. The crosswalk table is updated accordingly.

Validate Item type. The Program Module validates that the CFS Item Type values displayed in Appendix B are present and active in CFS. The crosswalk table is updated accordingly.

Validate Commodity Code. The Program Module validates that the CFS Commodity Code values displayed in Appendix C are present and active in CFS. The crosswalk table is updated accordingly.

Validate Payment Office Code Code. The Program Module validates that the CFS Payment Office Code values displayed in Appendix D are present and active in CFS. The crosswalk table is updated accordingly.

Upon completion of the Validate Document Data Function, the Program Module selects the first **FIMA Document Number** eligible for conversion and the associated data elements from **CONVERT ACTDOC** and **CONVERT ACTDOC ACCOUNT**.

2.3 Business Rules for Validate Document Data Function

Document data must be active within CFS to be present on a CFS Transaction.

2.4 Input to Validate Document Data Function

The following FIMA document data crosswalks and CFS tables provide the input needed to execute the Validate Document Data Function.

- **ACCOUNTING CODE** (**ACCOUNTING_CODE**)
- **PAYMENT OFFICE CONTROL** (**PAYMENT_OFFICE_CONTROL**)
- **COMMODITY** (**COMMODITY**)
- **DOCUMENT MATCHING** (**DOCUMENT_MATCHING**)
- **PAYMENT DEFAULT DETAIL** (**PAYMENT_DEFAULT_DETAIL**)
- FIMA Document Matching Matrix. See Appendix A.
- FIMA Item Type Matrix. See Appendix B.
- FIMA Commodity Code Matrix. See Appendix C.
- FIMA Payment Office Code Matrix. See Appendix D.

2.5 Validate Document Data Function Processing

2.5.1 Validate Document Type

The Program Module verifies the **CFS Document Types** displayed on the Document Matching Matrix (Appendix A), are present and active in CFS.

2.5.1.1 Logic

CFS Document Types are present on the **ACCOUNTING CODE** table in CFS and the **Active Status** (**Active_status**) is 'Y'. The **Code Type** (**Code_type**) is 'OBLIG' and the **Code Value** (**Code_value**) is the **CFS Document Type** from the FIMA Document Matching Matrix.

CFS Document Types are present on the **PAYMENT DEFAULT DETAIL** table in CFS and the **Active Status** is 'Y'. An entry for each **CFS Document Type** from the FIMA Document Matching Matrix are present in the **Document Type** (**Document_type**) field on the table.

CFS Document Types are present on the **DOCUMENT MATCHING** table in CFS and the **Active Status** is 'Y'. An entry for each **CFS Document Type** from the FIMA Document Matching Matrix are present in the **Document Type** (Document_type) field on the table. In addition, the appropriate matching criteria is displayed. **CFS Document Types** identified as 2-Way match documents by the CFS Matching column on the FIMA Document Matching Matrix, have a **PO Flag** (**PO_flag**) set to 'Y'. **CFS Document Types** identified as No-Match by the CFS Matching column on the FIMA Document Matching Matrix have a **PO Flag** set to 'N'.

2.5.1.2 Output

Update the **Active Status Flag** (**Active_status_flag**) on the Crosswalk table to 'N' for each CFS Document Type, based on the following criteria:

- The CFS Document Type does not reflect an **Active Status** of 'Y' on the **ACCOUNTING CODE** table.
- The CFS Document Type does not reflect an **Active Status** of 'Y' on the **PAYMENT DEFAULT DETAIL** table.
- The CFS Document Type does not reflect an **Active Status** of 'Y' or reflects an incorrect **PO Flag** on the **DOCUMENT MATCHING** table.

Update the **Active Status Flag** (**Active_status_flag**) on the Crosswalk table to 'Y' for each CFS Document Type, based on the following criteria:

- The CFS Document Type does reflect an **Active Status** of 'Y' on the **ACCOUNTING CODE** table.
- The CFS Document Type does reflect an **Active Status** of 'Y' on the **PAYMENT DEFAULT DETAIL** table.
- The CFS Document Type does reflect an **Active Status** of 'Y' AND reflects a **PO Flag** of 'Y' on the **DOCUMENT MATCHING** table for a 2-Way match document.
- The CFS Document Type does reflect an **Active Status** of 'Y' AND reflects a **PO Flag** of 'N' on the **DOCUMENT MATCHING** table for a No-Match document.

2.5.2 Validate Item Type

The Program Module verifies the **CFS Item Types** displayed on the FIMA Item Type Matrix (Appendix B), are present and active in CFS.

2.5.2.1 Logic

CFS Item Types are present on the **ACCOUNTING CODE** table in CFS and the **Active Status** (**Active_status**) is 'Y'. The **Code Type** (**Code_type**) is 'CDITEM' and the **Code Value** (**Code_value**) is the **CFS Item Type** from the FIMA Item Type Matrix.

CFS Item Types from the FIMA Item Type Matrix are present on the **DOCUMENT MATCHING** table in CFS for each **CFS Document Type** from the FIMA Document Matching Matrix and the **Active Status** is 'Y'.

2.5.2.2 Output

Update the **Active Status Flag** (**Active_status_flag**) on the Crosswalk table to 'N' for each CFS Item Type, based on the following criteria:

- The CFS Item Type does not reflect an **Active Status** of 'Y' on the **ACCOUNTING CODE** table.
- The CFS Item Type from the FIMA Item Type Matrix is not present for every **CFS Document Type** from the FIMA Document Matching Matrix.
- The CFS Item Type does not reflect an **Active Status** of 'Y' on the **DOCUMENT MATCHING** table

Update the **Active Status Flag** (**Active_status_flag**) on the Crosswalk table to 'Y' for each CFS Item Type, based on the following criteria:

- The CFS Item Type does reflect an **Active Status** of 'Y' on the **ACCOUNTING CODE** table.
- The CFS Item Type from the FIMA Item Type Matrix is present for every **CFS Document Type** from the FIMA Document Matching Matrix and reflects an **Active Status** of 'Y' on the **DOCUMENT MATCHING** table.

2.5.3 Validate Commodity Code

The Program Module verifies the **CFS Commodity Codes** (**Commodity_code**) displayed on the FIMA Commodity Code Matrix (Appendix C), are present and active in CFS.

2.5.3.1 Logic

All CFS Commodity Codes are present on the COMMODITY table in CFS and the Active Status is 'Y'.

2.5.3.2 Output

Update the **Active Status Flag** (**Active_status_flag**) on the Crosswalk table to 'N' for each CFS Commodity Code, based on the following criteria:

- The CFS Commodity Code does not reflect an **Active Status** of 'Y' on the **COMMODITY** table.

Update the **Active Status Flag** (**Active_status_flag**) on the Crosswalk table to 'Y' for each CFS Commodity Code, based on the following criteria:

- The CFS Commodity Code does reflect an **Active Status** of 'Y' on the **Commodity** table.

2.5.4 Validate Payment Office Code

The Program Module verifies the **CFS Payment Office Code** (**Payment_office_code**) displayed on the FIMA Payment Office Code Matrix (Appendix D), are present and active in CFS.

2.5.4.1 Logic

All CFS Payment Office Codes are present on the **PAYMENT OFFICE CONTROL** table in CFS and the **Active Status** is 'Y'

2.5.4.2 Output

Update the **Active Status Flag** (**Active_status_flag**) on the Crosswalk table to 'N' for each CFS Payment Office Code, based on the following criteria:

- The CFS Payment Office Code does not reflect an **Active Status** of 'Y' on the **PAYMENT OFFICE CONTROL** table.

Update the **Active Status Flag** (**Active_status_flag**) on the Crosswalk table to 'Y' for each CFS Payment Office Code, based on the following criteria:

- The CFS Payment Office Code does reflect an **Active Status** of 'Y' on the **PAYMENT OFFICE CONTROL** table.

3 Crosswalk Document Data Function

3.1 Purpose of Crosswalk Document Data Function

The purpose of this Function is to ensure that the data present for a given **FIMA Document Number** is sufficient to populate the derived fields in CFS. If a data element is encountered that has not been defined or is unrecognizable, the Program Module will update the **Error Flag** for the document on **CONVERT ACTDOC** and will select the next available **FIMA Document Number** for conversion.

3.2 Process Flow of Crosswalk Document Data Function

This section describes an overview of the major steps needed to validate the document data for each document to be converted from FIMA to CFS. The document data validation is illustrated in Figure 3.

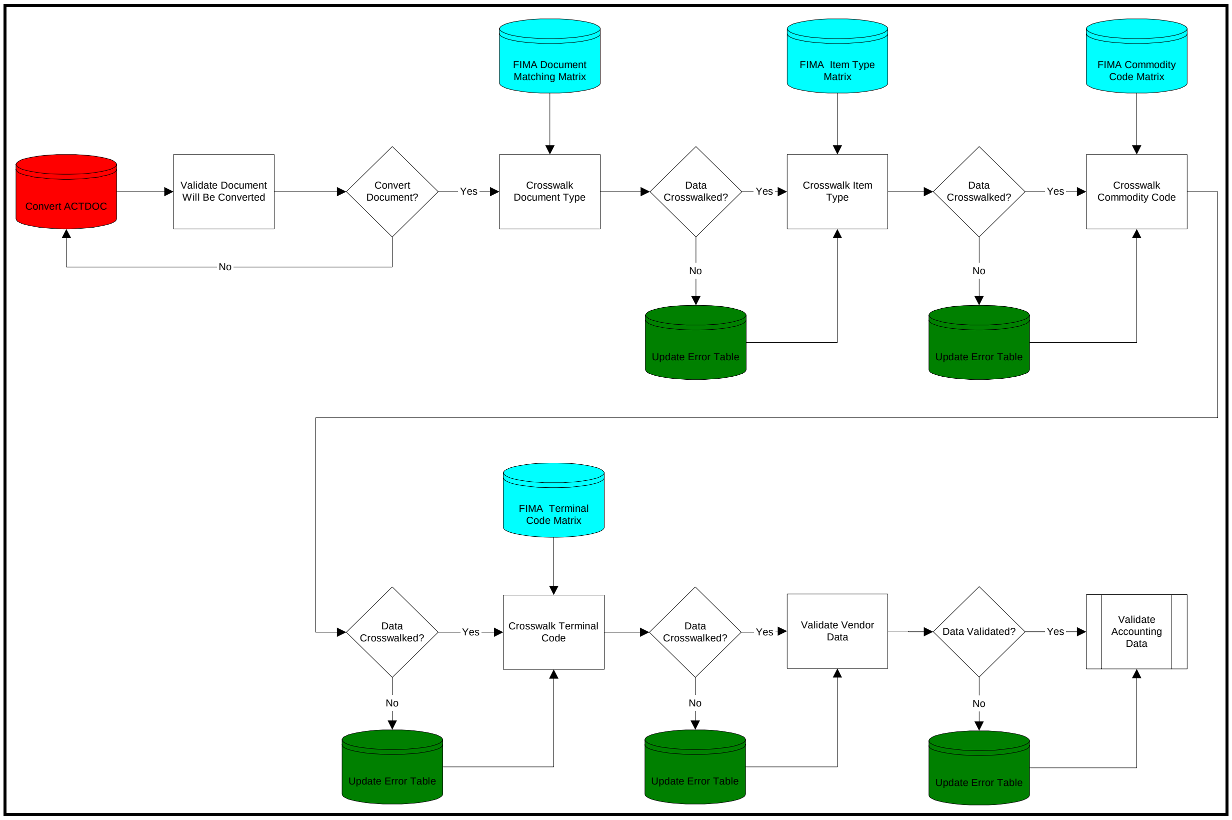

Figure 3. Crosswalk Document Data Process Flow

Validate Document Will Be Converted. Once the document data has been validated, the Program Module selects the first **FIMA Document Number** eligible for conversion and all the associated data elements from **CONVERT ACTDOC** (**CONV_ACTDOC** and **CONV_ACTDOC_ACCOUNT**). Any document that has been reconciled with CFS or has been successfully converted in a previous attempt is not processed during the current run of the Program Module.

Convert Document? If the FIMA Document Number is not eligible for conversion, the program will look to select the next FIMA **Document Number** from **CONVERT ACTDOC** and validate the document will be converted. If the

FIMA Document Number is identified as eligible for conversion, the **Document Sequence** (**Document_sequence**) numbers for each **FIMA Document Number** are ordered by **Fiscal Year** (**Fiscal_year**) to simplify insertion into CFS tables.

Crosswalk Document Type. **FIMA Document Numbers** identified as candidates for conversion will continue through the Crosswalk Document Data Function. The Crosswalk Document Type Sub-Function ensures that the FIMA Document Type can be crosswalked to a CFS value via the FIMA Document Matching Matrix.

Data Crosswalked? If the data is successfully crosswalked, the Program Module continues to the subsequent Sub-Function. Any **FIMA Document Number** with data that is not successfully crosswalked will be marked with an error flag and will appear on the error report produced by the Program Module. The Program Module will continue attempting to crosswalk data so errors can be identified at one time.

Crosswalk Item Type. The Crosswalk Item Type Sub-Function ensures that the **Major Object Class** and **Minor Object Class** can be crosswalked to a CFS value via the FIMA Item Type Matrix.

Data Crosswalked? Same as above Data Crosswalked? Sub-Function.

Crosswalk Commodity Code. The Crosswalk Commodity Code Sub-Function ensures that the **FIMA Document Type** can be crosswalked to a CFS value via the FIMA Commodity Code Matrix.

Data Crosswalked? Same as above Data Crosswalked? Sub-Function.

Crosswalk Payment Office Code. The Crosswalk Payment Office Code Sub-Function ensures that the **FIMA Terminal Code** can be crosswalked to a CFS value via the FIMA Payment Office Code Matrix.

Data Crosswalked? Same as above Data Crosswalked? Sub-Function.

Validate Vendor Data. The Validate Vendor Data Sub-Function ensures that the **Vendor Number** and **Vendor Id** can be validated in CFS.

Data Validated? If the data is successfully validated, the Program Module continues to the Validate Accounting Data Function. Any **FIMA Document**

Number with vendor data that is not successfully validated will be marked with an error flag and will appear on the error report produced by the Program Module. The Program Module continues to the Validate Accounting Data Function.

3.3 Input to Crosswalk Document Data Function

The following FIMA document data crosswalks, CFS tables, and **CONVERT ACTDOC** tables provide the input needed to execute the Crosswalk Document Data Function:

FIMA Document Matching Matrix. See Appendix A.
FIMA Item Type Matrix. See Appendix B.
FIMA Commodity Code Matrix. See Appendix C.
FIMA Payment Office Code Matrix. See Appendix D.
VENDOR DETAIL
CONVERT ACTDOC
CONVERT ACTDOC ACCOUNT

3.4 Crosswalk Document Data Function Processing

3.4.1 Validate Document Will Be Converted

The Program Module verifies only **FIMA Document Numbers** that have not yet been successfully converted, are considered eligible for conversion.

3.4.1.1 Logic

Select all the data elements in **CONVERT ACTDOC** and **CONVERT ACTDOC ACCOUNT** where the **Conversion Status Flag** and **Error Flag** are ‘N’ and where the FIMA Document Type does not equal '04'.

3.4.2 Crosswalk Document Type

The Program Module verifies the **FIMA Document Type** of the current FIMA **Document Number** is mapped in the Document Matching Matrix (Appendix A).

3.4.2.1 Logic

Verify the **FIMA Document Type** is present and has an **Active Status Flag** of ‘Y’ in the Crosswalk table.

3.4.2.2 Error Handling

If the **FIMA Document Type** is not present and active in the Crosswalk table, write an 'Invalid Document Type Error' to the **ERROR CONVERSION** table (**CONV_ERRORS**) and update the **Error Flag** and the **CFS Document Type Error** (**CFS_Document_Type_Error**) on **CONVERT ACTDOC** to 'Y'.

3.4.3 Crosswalk Item Type

The Program Module verifies the **Major Object Class** and **Minor Object Class** of the current FIMA Document Number is mapped to an active Item Type in the FIMA Item Type Matrix (Appendix B).

3.4.3.1 Logic

Verify that the **Major Object Class** and **Minor Object Class** combination of the current FIMA Document Number maps to an Item Type that has an **Active Status Flag** of ‘Y’ in the Crosswalk table.

3.4.3.2 Error Handling

If the **Major Object Class** and **Minor Object Class** combination is not mapped to an active **Item Type**, write an 'Invalid Item Type Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'.

3.4.4 Crosswalk Commodity Code

The Program Module verifies the **FIMA Document Type** of the current document is mapped in the Commodity Code Matrix (Appendix C).

3.4.4.1 Logic

Verify the **FIMA Document Type** maps to a **Commodity Code** that has an **Active Status Flag** of ‘Y’ in the Crosswalk table.

3.4.4.2 Error Handling

If the **FIMA Document Type** is not mapped to a **Commodity Code**, write an 'Invalid Commodity Code Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'.

3.4.5 Validate Payment Office Code

The Program Module verifies the **Terminal Code** of the current document is mapped in the Payment Office Code Matrix (Appendix D).

3.4.5.1 Logic

Verify the **FIMA Terminal Code** maps to a corresponding **Payment Office Code** that has an **Active Status Flag** of ‘Y’ in the Crosswalk table.

3.4.5.2 Error Handling

If the **FIMA Terminal Code** is not mapped to a **CFS Payment Office Code**, write an 'Invalid Payment Office Code Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'.

3.4.6 Validate Vendor Information

The Program Module verifies the **Vendor Number** and **Vendor ID** of the current document is present, active, and the correct **Address Type** in the CFS Vendor File.

3.4.6.1 Logic

Verify that the **Vendor Number** and **Vendor Id** from **CONVERT ACTDOC** has an Active Status of ‘Y’ on **VENDOR DETAIL**.

Verify that the **Vendor Number** and **Vendor Id** combination reflects an **Address Type** of 'PURCH' on **VENDOR DETAIL**.

3.4.6.2 Error Handling

If the **Vendor Number** and **Vendor id** are not present or active on **VENDOR DETAIL**, write an 'Invalid Vendor Number Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'.

If the **Vendor Id** is not of **Address Type** 'PURCH' on **VENDOR DETAIL**, write an 'Invalid Vendor Address Type Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'.

4 Validate Accounting Data Function

Once the Program Module completes processing the Crosswalk Document Data Function, the Validate Accounting Data Function is executed. This Function is executed irrespective of any errors encountered in the previous Function.

4.1 Purpose of Validate Accounting Data Function

The Program Module will call the Validate Accounting Data Function described in the Prepare FIMA ACTDOC for Open Document Conversion Program Module to ensure the accounting data (CFS ACCS values) inserted in CFS are valid.

Please reference the Prepare FIMA ACTDOC for Open Document Conversion Program Module for information on this Function.

5 Record Purchase Order Function

After the Validate Accounting Data Function, the Program Module assesses whether to continue processing the current FIMA Document Number. If an error is encountered during either the Crosswalk Document Data Function or the Validate Accounting Data Function (i.e. the **Error Flag** is set to 'Y' on **CONVERT ACTDOC**), the Program Module selects another FIMA Document Number from **CONVERT ACTDOC** and calls the Crosswalk Document Data Function.

If no error is encountered during the previous two Functions, the Program Module executes the Record Purchase Order Function.

5.1 Purpose of Record Purchase Order Function

FIMA Document Types identified as 2-Way match documents must be processed by the Record Purchase Order Function. The Record Purchase Order Function identifies how data from FIMA will be used to derive and default values needed to produce a Purchase Order in CFS.

5.2 Process Flow of Record Purchase Order Function

This section describes an overview of the major steps needed to convert 2-Way match documents from FIMA to CFS. The process to record a Purchase Order in CFS is illustrated in Figure 4.

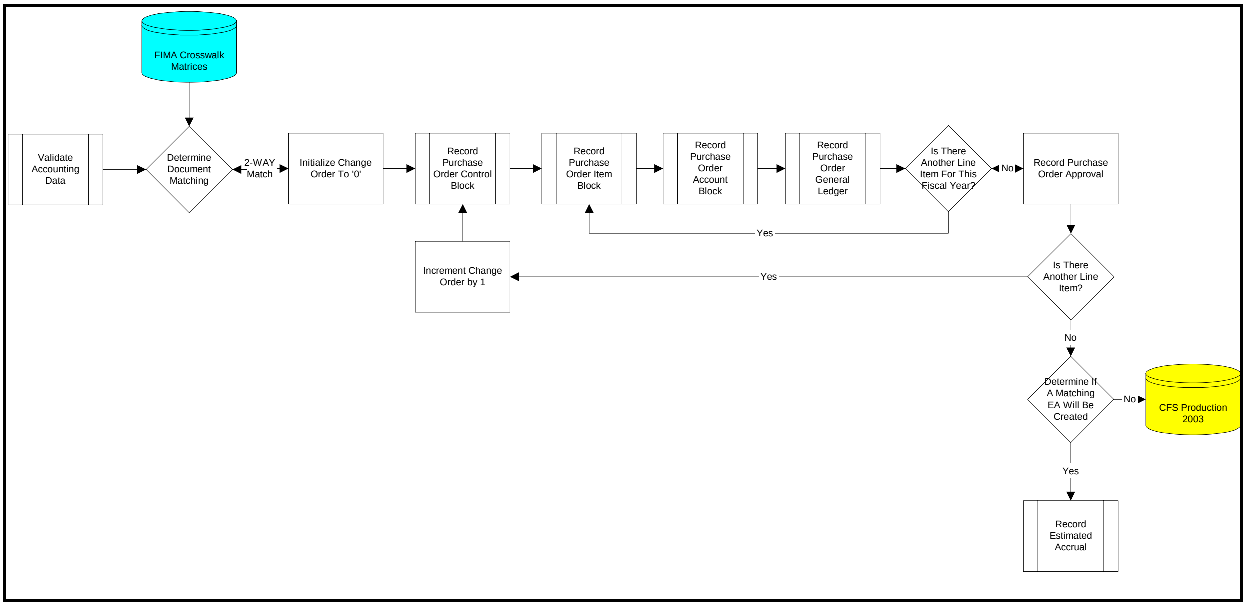
FIMA Crosswalk Matrices
Validate Accounting Data
Determine Document Matching
2-WAY Match
Initialize Change Order To '0'
Record Purchase Order Control Block
Record Purchase Order Item Block
Record Purchase Order Account Block
Record Purchase Order General Ledger
Is There Another Line Item For This Fiscal Year?
No
Record Purchase Order Approval
Yes
Increment Change Order by 1
Yes
Is There Another Line Item?
No
Determine If A Matching EA Will Be Created
No
CFS Production 2003
Yes
Record Estimated Accrual

Figure 4. Record Purchase Order Process Flow

Determine Document Matching. Section 1.3 Process Flow For The Module provides a description of this Sub-Function.

Initialize Change Order To '0'. The Program Module begins by initializing the Change Order counter to zero.

Record Purchase Order Control Block. The Program Module creates the Purchase Order Control Block for a given Fiscal Year.

Record Purchase Order Item Block. The Program Module creates the Purchase Order Item Block for each line item with the same Fiscal Year.

Record Purchase Order Account Block. The Program Module creates the Purchase Order Account for each Purchase Order Item Block created in the previous Sub-Function.

Record Purchase Order General Ledger. The Program Module creates the CFS General Ledger entries for each Purchase Order Account Block created in the previous Sub-Function.

Is There Another Line Item For This Fiscal Year? If another line item exist for the same Fiscal Year as the previously processed line item, the Program Module needs only to create a Purchase Order Item Block, Purchase Order Account Block, and Purchase Order General Ledger entries the line item. If no line item exists for the same Fiscal Year the Program Module continues to the Record Purchase Order Approval Sub-Function.

Record Purchase Order Approval. The Program Module creates the Purchase Order Approval and calls the Is There Another Line Item Sub-Function upon completion.

Is There Another Line Item? If an unconverted line item exists with a different Fiscal Year, the Program Module calls the Increment Change Order By 1 Sub-Function. If no unconverted line item exists (i.e. all line items have been converted), the Program Module calls the Determine If A Matching EA Will Be Created Sub-Function.

Increment Change Order by 1. The Program Module increments the Change Order counter by 1, and calls the Record Purchase Order Control Block Sub-Function.

Determine If A Matching EA Will Be Created. The Program Module determines whether to call the Record Estimated Accrual Function. If the Record Estimated Accrual Function is not called, the data created in the Record Purchase Order Function is moved to CFS Production 2003 and the Program Module selects the next FIMA Document Number eligible for conversion.

5.3 Record Purchase Order Function Processing

5.3.1 Initialize Change Order To ‘0'

The Program Module begins by initializing the Change Order counter to zero.

5.3.1.1 Logic

Set **Change Order Number** equal to zero.

5.3.1.2 Output

The current **Change Order Number** is zero.

5.3.2 Record Purchase Order Control Block

For a each distinct **Fiscal Year**, a single control block entry is made.

The following sections explain how to obtain the values needed to populate the Purchase Order Control Block. Because the derived values differ based on the **Change Order Number**, this Function is broken into initial control block (**Change Order Number** equals zero) and change order control block (**Change Order Number** is greater than zero).

5.3.2.1 Logic

Table 1 provides a field by field description of how to obtain the values to populate **PO CONTROL**. The Field Name column contains every field name present on **PO CONTROL**. The Source For Initial **PO CONTROL** column contains information for determining how the corresponding field name is populated when the **Change Order Number** is equal to zero. The Source For Change Order PO CONTROL contains information for determining how the corresponding field name is populated when the **Change Order Number** is greater than zero.

Table 1. PO CONTROL

Field Name	Source For Initial PO CONTROL	Source For The Change Order PO CONTROL
Document_no	select Sequence_no + 1 from Maxseqnos where Table_name equals ‘ PO_CONTROL_DOC’	current Document_no
Change_order_no	0	current Change_order_no value + 1
Release_no	0	PO_CONTROL.Release_no
Bureau_code	CONVERT_ACTDOC_ACCOUNT.Bureau_code	select Bureau_code from PO_CONTROL PO where PO.Document_no equals current Document_no
Display_fiscal_year	Last 2 digits of CONVERT_ACTDOC.Fiscal_year	Last 2 digits CONVERT_ACTDOC.Fiscal_year

Table 1. PO CONTROL		
Field Name	Source For Initial PO CONTROL	Source For The Change Order PO CONTROL
Fiscal_year	CONVERT_ACTDOC.Fiscal_year	CONVERT_ACTDOC.Fiscal_year
Acknowledged_flag	N	N
AP_Status	O	O
Approved_by	Null	Null
Approved_date	09/30/2002	09/30/2002
Approved_emp_no	0	0
Approved_flag	Y	Y
Begin_date	09/30/2002	09/30/2002
Change_order_flag	N	Y
Confirmed_flag	N	N
Confirmed_with	Null	Null
Contact	select VC.Contact from VENDOR_CONTROL VC, VENDOR_DETAIL VD where VC.Vendor_no equals CONVERT_ACTDOC.Vendor_no and VD.Vendor_id equals CA.Vendor_id and VD.Address_type equals 'PURCH' and VD.Active_status equals 'Y'	select Contact from PO_CONTROL where Document_no equals the current document and Change_order_no equals current Change_order_no minus 1.

Table 1. PO CONTROL		
Field Name	Source For Initial PO CONTROL	Source For The Change Order PO CONTROL
Country_code	select VD.Country_code (see Contact source information)	select Country_code (see Contact source information)
Discount_amount1	select VC.Discount_amount1 (see Contact source information)	select VC.Discount_amount1 (see Contact source information)
Discount_amount2	select VC.Discount_amount2 (see Contact source information)	select VC.Discount_amount2 (see Contact source information)
Discount_days1	select VC.Discount_days1 (see Contact source information)	select VC.Discount_days1 (see Contact source information)
Discount_days2	select VC.Discount_days1 (see Contact source information)	select VC.Discount_days1 (see Contact source information)
Discount_flag1	select VC.Discount_flag1 (see Contact source information)	select VC.Discount_flag1 (see Contact source information)
Discount_flag2	select VC.Discount_flag2 (see Contact source information)	select VC.Discount_flag2 (see Contact source information)
Document_Date	09/30/2002	09/30/2002
Document_status	OPEN	OPEN
Document_type	current Document_type	PO_CONTROL.Document_type
End_date	Null	Null
Gl_end_date	09/30/2002	09/30/2002
Exchange_rate	1	1

Table 1. PO CONTROL		
Field Name	Source For Initial PO CONTROL	Source For The Change Order PO CONTROL
Fixed_price_flag	N	N
Fob_point	DESTIN	DESTIN
Freight_terms	Null	Null
Future_dated	N	N
Interface_option_code	Null	Null
Interface_document_no	Null	Null
Net_days1	select VC.Net_days1 (see Contact source information)	select VC.Net_days1 (see Contact source information)
Net_days2	select VC.Net_days2 (see Contact source information)	select VC.Net_days2 (see Contact source information)
Not_to_exceed_amount	Null	Null
Notes	"This document has been converted from FIMA. To view the document in FIMA, please use the CFS Source Reference Number as the FIMA Document Number."	"This document has been converted from FIMA. To view the document in FIMA, please use the CFS Source Reference Number as the FIMA Document Number."
Placed_by	See employee data issue in section 10.	select Placed_by (see Contact source information)
Placed_date	09/30/2002	09/30/2002

Table 1. PO CONTROL		
Field Name	Source For Initial PO CONTROL	Source For The Change Order PO CONTROL
Placed_emp_no	See employee data issue in section 10.	select Places_emp_no from PO_CONTROL PO, CONVERT_ACTDOC CA where PO.Document_no equals CA.Cfs_document_number
Placed_flag	Y	Y
Print_flag	Y	N
Reference_no	CONVERT_ACTDOC.Document_number	CONVERT_ACTDOC.Document_number
Release_flag	N	N
Rt_status	O	O
Shipto_address1	Null	Null
Shipto_address2	Null	Null
Shipto_address3	Null	Null
Shipto_address4	Null	Null
Shipto_city	Null	Null
Shipto_code	TPR	TPR
Shipto_country	Null	Null
Shipto_name	Third Party	Third Party
Shipto_state	Null	Null
Shipto_zip_code	Null	Null

Table 1. PO CONTROL		
Field Name	Source For Initial PO CONTROL	Source For The Change Order PO CONTROL
Shipvia_code	Null	Null
Status_date	09/30/2002	09/30/2002
To_print_flag	N	N
Trans_no	select Sequence_no + 1 from Maxseqnos where Table_name equals ‘PO_CONTROL_TRANS’	select Sequence_no + 1 from Maxseqnos where Table_name equals ‘PO_CONTROL_TRANS’
Vendor_id	CONVERT_ACTDOC.Vendor_id	CONVERT_ACTDOC.Vendor_id
Vendor_no	CONVERT_ACTDOC.Vendor_no	CONVERT_ACTDOC.Vendor_no
One_to_one	N	N
Contracting_office_code	XX	XX
Award_type	Null	Null
Type_entry	Null	A
Order_type	Null	Null
Order_against	Null	Null
Foreign_flag	Null	Null
Cont_officer_no	Null	Null

Table 1. PO CONTROL		
Field Name	Source For Initial PO CONTROL	Source For The Change Order PO CONTROL
Contractor_type	Null	Null
Contract_type	Null	Null
Contract_kind	Null	Null
Ext_competition	Null	Null
Source	Null	Null
Multi_year	Null	Null
Preference_program	Null	Null
Ppp_state	Null	Null
Ppp_city	Null	Null
Est_contract_cost	Null	Null
Est_contract_date	Null	Null
Rpt_agency	Null	Null
Rpt_period	Null	Null
Cotr	Null	Null

Table 1. PO CONTROL		
Field Name	Source For Initial PO CONTROL	Source For The Change Order PO CONTROL
Ready_req_date	Null	Null
Award_date	Null	Null
Advisory_code	Null	Null
Dollar_associated	Null	Null
Bid_foreign	Null	Null
Buy_american	Null	Null
Country_mfgr	Null	Null
Competition_auth	Null	Null
Offers_received	Null	Null
Subcontracting_plan	Null	Null
Solicitation_proc	Null	Null
Labor_statutes	Null	Null
Cica	Null	Null
Dao	Null	Null

Table 1. PO CONTROL		
Field Name	Source For Initial PO CONTROL	Source For The Change Order PO CONTROL
Synopsis	Null	Null
Tariff	Null	Null
All_required_flag	Null	Null
Sic_code	Null	Null
Product_code	Null	Null
User_name	CONVERSION	CONVERSION
Modification_date	sysdate	sysdate
Device_name	CONVERSION	CONVERSION
Form_type	Null	Null
Originating_office_code	NA	NA
Comm_item	Null	Null
Rga	Null	Null
Pmat_save	Null	Null
Pmat_save_cat	Null	Null
Pmat_complex	Null	Null

Table 1. PO CONTROL

Field Name	Source For Initial PO CONTROL	Source For The Change Order PO CONTROL
Req_doc_no	Null	Null
Feeder_sys_no	Text 'Originating FY ' CONVERT_ACTDOC.Min_fiscal_year	Null
Rpt_override	Null	Null
Est_project_amount	Null	Null
Contracting_office_control	Null	Null
Note_trans_no	Null	Null
Reported_fein_no	Null	Null
Minority_owned_flag	Null	Null
Woman_owned_flag	Null	Null
Review_flag	Null	Null
Discount_applied	Null	Null
Lead_type	Null	Null
Po_amount	select sum (Undelivered_orders and Unpaid_accrued_exp) from CONVERT_ACTDOC where Fiscal_year is the same across line items	sum: 1) select Undelivered_orders and Unpaid_accrued_exp from CONVERT ACTDOC where Fiscal_year equals Fiscal_year of the current Document_Sequence 2) PO_Amount of last change_order_no

Table 1. PO CONTROL		
Field Name	Source For Initial PO CONTROL	Source For The Change Order PO CONTROL
Rqt_agency	Null	Null
Pbsc	Null	Null
Hubzone_sml_bus_concern_flag	Null	Null
Hubzone_prog_flag	Null	Null
Sml_disadv_bus_prog_flag	Null	Null
Hubzone_price_eval	Null	Null
Sml_disadv_bus_price_eval	Null	Null
Accrual_status	O	O
Service_disabled_vet_flag	Null	Null
Vetern_owned_flag	Null	Null

5.3.2.2 *Output*

Upon a successful completion of the Purchase Order Control Block Function, the Program Module:

- Inserts to **PO CONTROL**.
- Sets the **CFS Document Number** field on **CONVERT ACTDOC ACCOUNT** equal to the **Document Number** (Document_no) value from the recently created **PO CONTROL** record.

5.3.2.3 *Error Handling*

If an error is encountered while inserting into **PO CONTROL**, write an 'Unable to update **PO CONTROL** Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'. In addition, the Program Module aborts converting the current document and removes previously inserted data for the current document.

5.3.3 Record Purchase Order Item Block

The Program Module records an entry to **PO ITEM** for each record on **CONVERT ACTDOC ACCOUNT** with the same **Fiscal Year**.

The following Sub-Functions detail how to populate the Purchase Order Item table. Because the derived values differ based on the change order number this Sub-Function is broken into initial item block (**Change Order Number** equals zero) and change order item block (**Change Order Number** is greater than zero). The relationship between the **PO ITEM** table and the **PO CONTROL** table is the **Document Number**.

5.3.3.1 Logic

Table 2 provides a field by field description of how to obtain the values to populate **PO ITEM**. The Field Name column contains the fields present on **PO ITEM**. The Source For Initial **PO ITEM** column contains information for determining how the corresponding field name is populated when the **Change Order Number** is equal to zero. The Source For Change Order **PO ITEM** contains information for determining how the corresponding field name is populated when the **Change Order Number** is greater than zero.

When the **Change Order Number** is greater than zero, the Program Module must perform two tasks to complete the Record Purchase Order Item Block Function:

- Copy the data from previously created **PO ITEM** records with specific field values modified;
- Create new **PO ITEM** records that have the same **Fiscal Year** value as the Fiscal Year populated in the current Purchase Order Control Block.

Table 2 includes terminology to help identify the specific field values from copied **PO ITEM** records. Beneath the Source For Change Order **PO ITEM** column, field values that must be modified for copied records are designated with the term '*Copied:*' followed by the code necessary to acquire the modified value. Copied data in fields without this term can remain unchanged in the copied **PO ITEM** record. The copied data comes from the **PO ITEM** record with the current **Document Number** and where the **Change Order Number** is the current **Change Order Number** minus one.

For fields in which the copied data remains unchanged, the information beneath the Source For Change Order **PO ITEM** column identifies how to determine the corresponding field value for new **PO ITEM** records.

For fields in which the copied data is modified (i.e. the term '*Copied:*' is present), the information pertaining to new **PO ITEM** records is identified by the term '*New:*'.

Table 2. PO ITEM

Field Name	Source For Initial PO ITEM	Source For Change Order PO ITEM
Change_order_no	current PO_CONTROL. Change_order_no	*New:* current PO_CONTROL. Change_order_no *Copied:* current PO_CONTROL.Change_order_no
Document_no	current PO_CONTROL.Document_no	current PO_CONTROL.Document_no
Item_no	This value begins with one and is incremented by one for each new PO ITEM entry. Initialized to one for each unique Document_no.	Take the greatest Item_no from previous PO ITEM, and add one for each additional new PO ITEM entry.
Release_no	PO_CONTROL.Release_no	PO_CONTROL.Release_no
Trans_no	PO_CONTROL.Trans_no	*New:* PO_CONTROL.Trans_no *Copied:* PO_CONTROL.Trans_no
Ac_flag	Null	Null
Acceptance_days	select Return_days from PROMPT_PAY where Prompt_pay_type is current Prompt_pay_type	select Return_days from PROMPT_PAY where Prompt_pay_type is current Prompt_pay_type

Table 2. PO ITEM

Field Name	Source For Initial PO ITEM	Source For Change Order PO ITEM
Ain	Null	Null
Asset_category	Null	Null
Asset_indicator	Null	Null
Ap_flag	Null	Null
Commodity_code	current Commodity_code	current Commodity_code
Default_matching_flag	Y	Y
Delivery_date	09/30/2002	09/30/2002
Deliver_to	PO_CONTROL.Placed_by	PO_CONTROL.Placed_by
Document_status	OPEN	OPEN
Dpoc_emp_no	PO_CONTROL.Placed_emp_no	PO_CONTROL.Placed_emp_no
Dpoc_location	select Code_descr from EMPLOYEE_CONTROL EC, ACCOUNTING_CODE AC where AC.Code_value equals EC.Location_no and AC.Code_type equals ‘EMPLOC’ and EC.Emp_no equals PO_CONTROL.Placed_emp_no	select Code_descr from EMPLOYEE_CONTROL EC, ACCOUNTING_CODE AC where AC.Code_value equals EC.Location_no and AC.Code_type equals ‘EMPLOC’ and EC.Emp_no equals PO_CONTROL.Placed_emp_no
Dpoc_phone_no	select Work_phone_no from EMPLOYEE_CONTROL where EC.Emp_no equals PO_CONTROL.Placed_emp_no	select Work_phone_no from EMPLOYEE_CONTROL where EC.Emp_no equals PO_CONTROL.Placed_emp_no
Due_date	09/30/2002	09/30/2002

Table 2. PO ITEM		
Field Name	Source For Initial PO ITEM	Source For Change Order PO ITEM
Emp_no	Null	Null
Holdback_amount	Null	Null
Holdback_type	N	N
Inventory_qty	Null	Null
Inventory_unit	Null	Null
Item_type	current Item_type	current Item_type
Old_change_order_no	Null	*New:* Null *Copied:* PO_CONTROL.Change_order_no minus 1
Ordered_qty	1	1
Ordered_unit	EA	EA
Part_No	CONVERT_ACTDOC_ACCOUNT.Organization_code - CONVERT_ACTDOC_ACCOUNT.Phase_code - CONVERT_ACTDOC_ACCOUNT.Task_code	CONVERT_ACTDOC_ACCOUNT.Organization_code - CONVERT_ACTDOC_ACCOUNT.Phase_code - CONVERT_ACTDOC_ACCOUNT.Task_code
Payment_office_code	current Payment_office_code	current Payment_office_code
Product_service_code	Null	Null
Qty_recd	Null	Null
Req_flag	N	N
Rt_status	O	O
Rt_flag	Null	Null
Status_date	09/30/2002	09/30/2002

Table 2. PO ITEM		
Field Name	Source For Initial PO ITEM	Source For Change Order PO ITEM
Stock_no	CONVERT_ ACTDOC_ACCOUNT.Fiscal_ year	CONVERT_ ACTDOC_ACCOUNT.Fiscal_ year
Unitprice	CONVERT_ACTDOC_ACCOUNT.(Undelivered_orders + Unpaid_accrued_exp)	CONVERT_ACTDOC_ACCOUNT.(Undelivered_orders + Unpaid_accrued_exp)
User_name	CONVERSION	CONVERSION
Modification_date	sysdate	sysdate
Device_name	CONVERSION	CONVERSION
Discount_per_unit	Null	Null
Discount_percent	Null	Null
Original_unitprice	Null	Null
Trade_in_qty	Null	Null
Trade_in_price	Null	Null
Trade_in_descr	Null	Null
Ri_method	D	D
Amount_recd	Null	Null
Accrual_status	O	O
Accrual_amount	Null	Null
Accrual_qty	Null	Null

5.3.3.2 Output

Upon a successful completion of the Purchase Order Item Block Function, the Program Module:

- Inserts to **PO ITEM**.

- Sets the **CFS Item Number** field on **CONVERT ACTDOC ACCOUNT** equal to the **Item Number** (Item_no) value from the recently created **PO ITEM** record.

For copied PO ITEM records the following field(s) on **PO ITEM** are modified:

- Update the **Change Order Number** field to the current **Change Order Number** value.
- Update the **Old Change Order Number** (**Old_change_order_no**) to the **Change Order Number** value minus one.
- Update the **Transaction Number** to the current **Transaction Number** from the recently created **PO ITEM** record.

5.3.3.3 Error Handling

If an error is encountered while inserting to **PO ITEM**, write an 'Unable to update PO ITEM Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'. In addition, the Program Module aborts converting the current document and removes previously inserted data for the current document.

5.3.4 Record Purchase Order Account Block

The Program Module records an entry to **PO ACCOUNT** for each entry to **PO ITEM**.

The following sub-functions detail how to obtain every value needed to populate the CFS Purchase Order Account tables. Because the derived values differ based on the change order number this function is broken into initial account block (**Change Order Number** equals zero) and change order account block (**Change Order Number** is greater than zero).

5.3.4.1 Logic

Table 3 provides a field by field description of how to obtain the values to populate **PO ACCOUNT**. The Field Name column contains the fields present on **PO ACCOUNT**. The Source For Initial **PO ACCOUNT** column contains information for determining how the corresponding field name is populated when the **Change Order Number** is equal to zero. The Source For Change Order **PO ACCOUNT** contains information for determining how the corresponding field name is populated when the **Change Order Number** is greater than zero.

When the **Change Order Number** is greater than zero, the Program Module must perform two tasks to complete the Record Purchase Order Account Block Function:

1. Copy the data from previously created **PO ACCOUNT** records with specific field values modified;
2. Create new **PO ACCOUNT** records that have the same **Fiscal Year** value as the **Fiscal Year** populated in the current Purchase Order Control Block.

Table 3 includes terminology to help identify the specific field values from copied **PO ACCOUNT** records. Beneath the Source For Change Order **PO ACCOUNT** column, field values that must be modified for copied records are designated with the term '*Copied:*' followed by the code necessary to acquire the modified value. Copied data in fields without this term can remain unchanged in the copied **PO ACCOUNT** record.

For fields in which the copied data remains unchanged, the information beneath the Source For Change Order **PO ACCOUNT** column identifies how to determine the corresponding field value for new **PO ACCOUNT** records.

For fields in which the copied data is modified (i.e. the term '*Copied:*' is present), the information pertaining to new **PO ACCOUNT** records is identified by the term '*New:*'.

Table 3. PO ACCOUNT

Field Name	Source For Initial PO ACCOUNT	Source For The Change Order PO Account
Change_order_no	PO_CONTROL.Change_order_no	*New:* current PO_CONTROL. Change_order_no *Copied:* current PO_CONTROL.Change_order
Document_no	PO_CONTROL.Document_no	PO_CONTROL.Document_no
Item_no	PO_ITEM.Item_no	PO_ITEM.Item_no
Line_No	1	1

Table 3. PO ACCOUNT

Field Name	Source For Initial PO ACCOUNT	Source For The Change Order PO Account
Release_no	PO_CONTROL.Release_no	PO_CONTROL.Release_no
Trans_no	PO_CONTROL.Trans_no	*New:* PO_CONTROL.Trans_no *Copied:* PO_CONTROL.Trans_no
Acceptance_emp_no	PO_CONTROL. Placed_emp_no	PO_CONTROL. Placed_emp_no
Amount	CONVERT_ACTDOC_ACCOUNT.(Undelivered_orders + Unpaid_accrued_exp)	CONVERT_ACTDOC_ACCOUNT.(Undelivered_orders + Unpaid_accrued_exp)
Ap_amount	Null	Null
Ap_qty	If Unpaid_accrued_exp equals 0, then Ap_qty equals Null Else Ap_qty equals 1	If Unpaid_accrued_exp equals 0, then Ap_qty equals Null Else Ap_qty equals 1
Ap_status	O	O
Direct_flag	select P.Direct_flag from PROJECT P, CONVERT_ACTDOC_ACCOUNT CAA where P.Project_code equals CAA. Project_code	select P.Direct_flag from PROJECT P, CONVERT_ACTDOC_ACCOUNT CAA where P.Project_code equals CAA. Project_code
Bureau_code	CONVERT_ACTDOC_ACCOUNT.Bureau_code	CONVERT_ACTDOC_ACCOUNT.Bureau_code
Fund_code	CONVERT_ACTDOC_ACCOUNT.Fund_code	CONVERT_ACTDOC_ACCOUNT.Fund_code
Org1_code	CONVERT_ACTDOC_ACCOUNT.Org1_code	CONVERT_ACTDOC_ACCOUNT.Org1_code
Org2_code	CONVERT_ACTDOC_ACCOUNT.Org2_code	CONVERT_ACTDOC_ACCOUNT.Org2_code

Table 3. PO ACCOUNT

Field Name	Source For Initial PO ACCOUNT	Source For The Change Order PO Account
Org3_code	CONVERT_ACTDOC_ACCOUNT.Org3_code	CONVERT_ACTDOC_ACCOUNT.Org3_code
Org4_code	CONVERT_ACTDOC_ACCOUNT.Org4_code	CONVERT_ACTDOC_ACCOUNT.Org4_code
Org5_code	CONVERT_ACTDOC_ACCOUNT.Org5_code	CONVERT_ACTDOC_ACCOUNT.Org5_code
Org6_code	CONVERT_ACTDOC_ACCOUNT.Org6_code	CONVERT_ACTDOC_ACCOUNT.Org6_code
Org7_code	CONVERT_ACTDOC_ACCOUNT.Org7_code	CONVERT_ACTDOC_ACCOUNT.Org7_code
Program1_code	CONVERT_ACTDOC_ACCOUNT.Program1_code	CONVERT_ACTDOC_ACCOUNT.Program1_code
Program2_code	CONVERT_ACTDOC_ACCOUNT.Program2_code	CONVERT_ACTDOC_ACCOUNT.Program2_code
Program3_code	CONVERT_ACTDOC_ACCOUNT.Program3_code	CONVERT_ACTDOC_ACCOUNT.Program3_code
Program4_code	CONVERT_ACTDOC_ACCOUNT.Program4_code	CONVERT_ACTDOC_ACCOUNT.Program4_code
Project_code	CONVERT_ACTDOC_ACCOUNT.Project_code	CONVERT_ACTDOC_ACCOUNT.Project_code
Task_code	CONVERT_ACTDOC_ACCOUNT.Task_code	CONVERT_ACTDOC_ACCOUNT.Task_code
Object1_code	CONVERT_ACTDOC_ACCOUNT Object1_code	CONVERT_ACTDOC_ACCOUNT Object1_code
Object2_code	CONVERT_ACTDOC_ACCOUNT Object2_code	CONVERT_ACTDOC_ACCOUNT Object2_code

Table 3. PO ACCOUNT		
Field Name	Source For Initial PO ACCOUNT	Source For The Change Order PO Account
Object3_code	CONVERT_ACTDOC_ACCOUNT Object3_code	CONVERT_ACTDOC_ACCOUNT Object3_code
Object4_code	CONVERT_ACTDOC_ACCOUNT Object4_code	CONVERT_ACTDOC_ACCOUNT Object4_code
User_define_accs	0	0
Req_bureau_code	Null	Null
Req_fund_code	Null	Null
Req_org1_code	Null	Null
Req_org2_code	Null	Null
Req_org3_code	Null	Null
Req_org4_code	Null	Null
Req_org5_code	Null	Null
Req_org6_code	Null	Null
Req_org7_code	Null	Null
Req_program1_code	Null	Null
Req_program2_code	Null	Null

Table 3. PO ACCOUNT		
Field Name	Source For Initial PO ACCOUNT	Source For The Change Order PO Account
Req_program3_code	Null	Null
Req_program4_code	Null	Null
Req_project_code	Null	Null
Req_task_code	Null	Null
Req_object1_code	Null	Null
Req_object2_code	Null	Null
Req_object3_code	Null	Null
Req_object4_code	Null	Null
Req_user_define_accs	Null	Null
Old_change_order_no	Null	*New:* Null *Copied:* PO_CONTROL.Change_order_no minus 1
Ordered_qty	1	1
Percent	100	100
Po_req_trans_no	Null	Null
Req_amount	Null	*New:* Null

Table 3. PO ACCOUNT		
Field Name	Source For Initial PO ACCOUNT	Source For The Change Order PO Account
		Copied: 0
Req_dept_no	Null	Null
Req_document_no	Null	Null
Req_document_type	Null	Null
Req_release_no	Null	Null
Req_item_type	Null	Null
Req_item_no	Null	Null
Req_line_no	Null	Null
Req_amount_placed	Null	*New:* Null *Copied:* (negative) PO_ACCOUNT.Amount
Req_trans_no	Null	Null
Ro_no	CAMS	CAMS
Rt_qty	0	0
Rt_status	O	O
Trans_code	To Be Determined	*New:* To Be Determined *Copied:* Null
Approved_by	PO_CONTROL.Placed_by	PO_CONTROL.Placed_by

Table 3. PO ACCOUNT		
Field Name	Source For Initial PO ACCOUNT	Source For The Change Order PO Account
Approved_date	09/30/2002	09/30/2002
Approved_emp_no	PO_CONTROL.Placed_emp_no	PO_CONTROL.Placed_emp_no
Approved_flag	Y	Y
Approved_title	select Title from EMPLOYEE_CONTROL where emp_no equals PO_CONTROL.Placed_emp_no	select Title from EMPLOYEE_CONTROL where emp_no equals PO_CONTROL.Placed_emp_no
Approved_work_phone_no	select Work_phone_no from EMPLOYEE_CONTROL where emp_no equals PO_CONTROL.Placed_emp_no	select Work_phone_no from EMPLOYEE_CONTROL where emp_no equals PO_CONTROL.Placed_emp_no
Override_by	Null	Null
Override_date	Null	Null
Override_emp_no	Null	Null
Override_flag	Null	Null
Override_title	Null	Null
Override_work_phone_no	Null	Null
User_name	CONVERSION	CONVERSION
Modification_date	sysdate	sysdate
Device_name	CONVERSION	CONVERSION
Changed_flag	I	*New:* I *Copied:* N
Rt_amount	Null	Null
Accrual_status	O	O
Accrual_amount	Null	Null
Accrual_qty	Null	Null

5.3.4.2 *Output*

Upon a successful completion of the Purchase Order Account Block Function, the Program Module:

- Inserts to **PO ACCOUNT**.

For copied **PO ACCOUNT** records the following field(s) on **PO ACCOUNT** are modified:

- Update the **Transaction Number** field to the **Transaction Number** of the current **PO CONTROL** record.
- Update the **Change Order Number** field to the current **Change Order Number** value.
- Update the **Old Change Order Number** (**Old_change_order_no**) to the **Change Order Number** value minus one.
- Update the **Requisition Amount** (**Req_amount**) to zero.
- Update the **Requisition Amount Placed** (**Req_amount_placed**) to a negative value of the **Amount** field.
- Update the **Transaction Code** (**Trans_code**) to Null.
- Update the **Changed Flag** (**Changed_flag**) to a value of 'N'

5.3.4.3 *Error Handling*

If an error is encountered while inserting into **PO ACCOUNT**, write an 'Unable to update **PO ACCOUNT** Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'. In addition, the Program Module aborts converting the current document and removes previously inserted data for the current document.

5.3.5 *Record Purchase Order General Ledger*

The Program Module records an entry to the CFS General Ledger after completion of the Record Purchase Order Account Block Function. In addition, the Program Module only records an entry to the CFS General Ledger for newly created **PO ACCOUNT** records. No entry to the CFS General Ledger occurs for copied **PO ACCOUNT** records.

5.3.5.1 *Logic*

Table 4 provides a field by field description of how to obtain the values to populate the CFS General Ledger (**TRIAL**). The Field Name column contains the fields present on **TRIAL**. The Source For Purchase Order **TRIAL** Table column contains information for determining how the corresponding field name is populated.

<table>
<tr><th colspan="2">Table 4. TRIAL For A Purchase Order Transaction</th></tr>
<tr><th>Field Name</th><th>Source For Purchase Order TRIAL Table</th></tr>
<tr><td>Gl_trans_no</td><td>Next sequence number (i.e. trial_seq.NEXTVAL)</td></tr>
<tr><td>Trans_date</td><td>09/30/2002</td></tr>
<tr><td>Gl_end_date</td><td>09/30/2002</td></tr>
<tr><td>Fiscal_quarter</td><td>4</td></tr>
<tr><td>Fiscal_year</td><td>2002</td></tr>
<tr><td>Fy_end_date</td><td>09/30/2002</td></tr>
<tr><td>Subsystem_code</td><td>OB</td></tr>
<tr><td>Trans_source</td><td>OBLIG</td></tr>
<tr><td>Trans_ref</td><td>'OBLIG-'||lpad(to_char(PO_CONTROL.Trans_no),8 , '0')
from PO_CONTROL</td></tr>
<tr><td>Trans_descr</td><td>'TC:'||PO_CONTROL.Trans_code||' - '||’OPEN DOCUMENT CONVERSION’ ||'</td></tr>
<tr><td>Trans_no</td><td>PO_CONTROL.Trans_no</td></tr>
<tr><td>Trans_type_flag</td><td>G</td></tr>
<tr><td>Direct_flag</td><td>PO_ACCOUNT.Direct_flag</td></tr>
<tr><td>Stat_unit_code</td><td>Null</td></tr>
<tr><td>Stat_unit_qty</td><td>Null</td></tr>
<tr><td>Emp_no</td><td>Null</td></tr>
<tr><td>Hourly_rate</td><td>Null</td></tr>
<tr><td>Debit_amount</td><td>When TRIAL.Account_no equals 4610:
PO_ACCOUNT.Amount

Else: Null</td></tr>
</table>

Table 4. TRIAL For A Purchase Order Transaction	
Field Name	**Source For Purchase Order TRIAL Table**
Credit_amount	When TRIAL.Account_no equals 4801: PO_ACCOUNT.Amount Else: Null
Budget_amount	When TRIAL.Account_no equals 4610: negative PO_ACCOUNT.Amount Else: 0
Fund_code_fiscal_year	CONVERT_ACTDOC_ACCOUNT.Fiscal_year
Fund_code	PO_ACCOUNT.Fund_code
Bureau_code	PO_ACCOUNT.Bureau_code
Org1_code	PO_ACCOUNT.Org1_code
Org2_code	PO_ACCOUNT.Org2_code
Org3_code	PO_ACCOUNT.Org3_code
Org4_code	PO_ACCOUNT.Org4_code
Org5_code	PO_ACCOUNT.Org5_code
Org6_code	PO_ACCOUNT.Org6_code
Org7_code	PO_ACCOUNT.Org7_code
Program1_code	PO_ACCOUNT.Program1_code
Program2_code	PO_ACCOUNT.Program2_code
Program3_code	PO_ACCOUNT.Program3_code
Program4_code	PO_ACCOUNT.Program4_code
Project_code	PO_ACCOUNT.Project_code
Task_code	PO_ACCOUNT.Task_code
Object1_code	PO_ACCOUNT.Object1_code

Table 4. TRIAL For A Purchase Order Transaction

Field Name	Source For Purchase Order TRIAL Table
Object2_code	PO_ACCOUNT.Object2_code
Object3_code	PO_ACCOUNT.Object3_code
Object4_code	PO_ACCOUNT.Object4_code
User_define_accs	PO_ACCOUNT.User_define_accs
Account_no	To Be Determined
Sub_account_no	To Be Determined
Full_account_no	PO_ACCOUNT.Bureau_code, ‘-‘, PO_ACCOUNT.Project, ‘-‘, PO_ACCOUNT.Task_code, ‘-‘, PO_ACCOUNT.Fund_Code, ‘-‘, PO_ACCOUNT.Program1_Code, ‘-‘, PO_ACCOUNT.Program2_Code, ‘-‘, PO_ACCOUNT.Program3_Code, ‘-‘, PO_ACCOUNT.Program4_Code, ‘-‘, PO_ACCOUNT.Org1_code, ‘-‘, PO_ACCOUNT.Org2_code, ‘-‘, PO_ACCOUNT.Org3_code, ‘-‘, PO_ACCOUNT.Org4_code, ‘-‘, PO_ACCOUNT.Org5_code, ‘-‘, PO_ACCOUNT.Org6_code, ‘-‘, PO_ACCOUNT.Org7_code, ‘-‘, PO_ACCOUNT.Object1_Code, ‘-‘, PO_ACCOUNT.Object2_Code, ‘-‘, PO_ACCOUNT.Object3_Code, ‘-‘, PO_ACCOUNT.Object4_Code, ‘000000', PO_ACCOUNT.Account_no, ‘-‘, PO_ACCOUNT. Sub_Account_No
Apport_category	A
Apportionment_no	Null
Advice_no	Null

Table 4. TRIAL For A Purchase Order Transaction

Field Name	Source For Purchase Order TRIAL Table
Avail_begin_date	Null
Avail_end_date	Null
Category_item_no	0
Begbal_flag	N
Document_type	PO_CONTROL.Document_type
Document_no	PO_CONTROL.Document_no
Document_date	PO_Control.Document_date
Change_order_no	PO_CONTROL.Change_order_no
Release_no	PO_CONTROL.Release_no
Vendor_no	PO_CONTROL.Vendor_no
Vendor_id	PO_CONTROL.Vendor_id
Customer_no	Null
Contact_no	Null
Reference_no	PO_CONTROL.Reference_no
Item_type	PO_ITEM.Item_type
Item_no	PO_ITEM.Item_no
Line_no	1
Original Fields Will Be Formatted If the Account_no equals ‘4801'	
Org_document_date	09/30/2002
Org_document_type	PO_CONTROL.Document_type
Org_document_no	PO_CONTROL.Document_no
Org_release_no	PO_CONTROL.Release_no
Org_item_type	PO_ITEM.Item_type

Table 4. TRIAL For A Purchase Order Transaction	
Field Name	Source For Purchase Order TRIAL Table
Org_item_no	PO_ITEM.Item_no
Org_line_no	1
User_name	CONVERSION
Modification_date	sysdate
Device_name	CONVERSION
Financing_flag	Null
Sf224_required	Null
Agency_location_code	Null
Cash_trans_type	Null
Accomplished_date	Null
Trial_id	Null (Updated via a trigger on trial)
Trans_code	PO_ACCOUNT.Trans_code
Reorg_token	Null
Account_no_digit_1	first digit of Account_no from TRIAL
Account_no_digit_2	second digit of Account_no from TRIAL

5.3.5.2 *Error Handling*

If an error is encountered while inserting into **TRIAL**, write an 'Unable to update **TRIAL** Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'. In addition, the Program Module aborts converting the current document and removes previously inserted data for the current document.

5.3.6 *Is There Another Line Item For This Fiscal Year*

If another line item exist for the same Fiscal Year as the previously processed line item, the Program Module needs only to create a Purchase Order Item Block, Purchase Order Account Block, and Purchase Order General Ledger entries the line item. If no line item exists for the same Fiscal Year the Program Module continues to the Record Purchase Order Approval Sub-Function.

5.3.6.1 *Logic*

The Program Modules increments the cursor to the next record from **CONVERT ACTDOC ACCOUNT**. If the Fiscal Year is the same as the previously processed Fiscal Year, the Program Module executes the Record Purchase Order Item Block Sub-Function and continues as outlined above (Record Purchase Order Item Block Sub-Function, Record Purchase Order Account Block Sub-Function, and Record Purchase Order General Ledger Sub-Function).

If the Fiscal Year is not the same as the previously processed Fiscal Year, the Program Module executes the Record Purchase Order Approval Sub-Function.

5.3.7 Record Purchase Order Approval

The Program Module records an entry to the Purchase Order Approval table (**PO APPROVAL**) after it is determined that no other line items exist for the current **Fiscal Year**.

5.3.7.1 Logic

Table 5 provides a field by field description of how to obtain the values to populate **PO Approval**. The Field Name column contains the fields present on **PO Approval**. The Source For PO Approval column contains information for determining how the corresponding field name is populated.

Table 5: Record PO APPROVAL Table

Field Name	Source For PO APPROVAL
Trans_no	PO_CONTROL.Trans_no
Approved_emp_no	PO_CONTROL.Placed_emp_no
Approved_by	PO_CONTROL.Placed_by
Approved_date	09/30/2002
Approved_flag	Y
Approved_title	PO_ACCOUNT.Approved_title
Approved_work_phone_no	PO_ACCOUNT.Approved_work_phone_no
Funds_cert_flag	Y
Notes	Null
Routing_priority	1
User_name	CONVERSION
Modification_date	sysdate
Device_name	CONVERSION

Table 5: Record PO APPROVAL Table	
Field Name	Source For PO APPROVAL
Document_sub_no	Null

5.3.7.2 Error Handling

If an error is encountered while inserting into **PO APPROVAL**, write an 'Unable to update **PO APPROVAL** Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'. In addition, the Program Module aborts converting the current document and removes previously inserted data for the current document.

5.3.8 Is There Another Line Item

If another line item exists, the Program Module executes the Increment Change Order By 1 Sub-Function. If no unprocessed line item exists, the Program Module executes the Determine If A Matching EA Will Be Created Sub-Function.

5.3.8.1 Logic

The Program Module increments the cursor to the next record from **CONVERT ACTDOC ACCOUNT**. If a record exists, the Program Module executes the Increment Change Order By 1 Sub-Function.

If no record record exists (i.e. the Program Module has processed all the line items in **CONVERT ACTDOC ACCOUNT**), the Program Module executes the Determine If A Matching EA Will Be Created Sub-Function.

5.3.9 Increment Change Order by 1

The Program Module begins by incrementing the Change Order counter by one, and executes the Record Purchase Order Control Block Sub-Function.

5.3.9.1 Logic

Add one to the **Change Order Number**.

5.3.10 Determine If A Matching EA Will Be Created

The Program Module determines whether to call the Record Estimated Accrual Function. If the Record Estimated Accrual Function is not called, the data created in the Record Purchase Order Function is moved to CFS Production 2003 and the Program Module selects the next FIMA Document Number eligible for conversion.

5.3.10.1 Logic

If the **Unpaid Accrued Expenditure** field for a **FIMA Document Number** is greater than zero for any line item on **CONVERT ACTDOC ACCOUNT**, the Program Module executes the Record Estimated Accrual Function to produce an Estimated Accrual for that line item record.

If no **Unpaid Accrued Expenditure** field is greater than zero the Program Module:

- Updates the **Conversion Status Flag** on **CONVERT ACTDOC** to reflect a value of 'Y'.
- Updates the **Conversion Date** (**Conv_date**) on **CONVERT ACTDOC** with the system date.
- Executes the Crosswalk Document Data Function to determine the next **FIMA Document Number** eligible for conversion.

6 Record Estimated Accrual Function

The Program Module can execute the Record Estimated Accrual Function via two different paths based on the document matching criteria:

1. If the selected document is identified as a 2-Way match document, the Determine If A Matching EA Will Be Created Sub-Function can execute the Record Estimated Accrual Function based on the logic in section 5.3.10.1. When the Record Estimated Accrual Function is executed in this manner, the Program Module adheres to the 2-Way match logic described in the Function.

2. OR, if the selected document is identified as a No-Match document, the Record Estimated Accrual Function is executed. When the Record Estimated Accrual Function is executed in this manner, the Program Module adheres to the No-Match logic described in the Function. Only documents with the **Conversion Status Flag** and **Error Flag** set to 'N' and **FIMA Document Type** does not equal '04' will be selected for this Function.

6.1 Purpose of Record Estimated Accrual Function

The Record Estimated Accrual Function identifies how data from FIMA will be used to derive and default values needed to produce an Estimated Accrual in CFS. The Estimated Accrual can be modified on the *Estimated Accrual Transaction Screen (PM050)*.

6.2 Process Flow of Record Estimated Accrual Function

This section describes an overview of the major steps needed to establish an Estimated Accrual document in CFS. The process is illustrated in Figure 4.

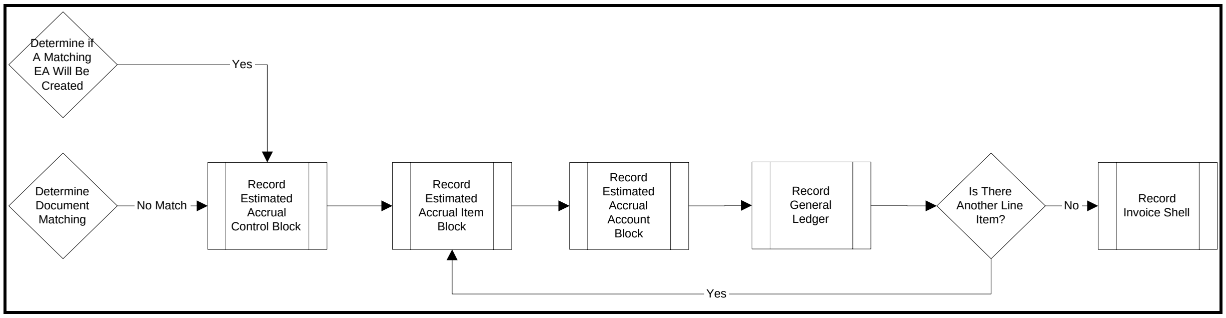
Determine if A Matching EA Will Be Created
Yes
Determine Document Matching
No Match
Record Estimated Accrual Control Block
Record Estimated Accrual Item Block
Record Estimated Accrual Account Block
Record General Ledger
Is There Another Line Item?
No
Record Invoice Shell
Yes

Figure 4. Record Estimated Accrual Process Flow

Determine If A Matching EA Will Be Created. This Sub-Function is the last step in the Record Purchase Order Function. The Program Module assesses whether it to execute the Record Estimated Accrual Function. If an Estimated Accrual is required based on the criteria from section 5.3.10.1, the Record Estimated Accrual Control Block Sub-Function is executed with the 2-Way match logic.

Determine Document Matching. After the Program Module executes the Validate Document Data Function, it determines the document matching criteria of the selected document. If the document is identified as a No-Match document, the Record Estimated Accrual Control Block Sub-Function is executed with the No-Match logic.

Record Estimated Accrual Control Block. The Program Module creates the Estimated Accrual Control Block based on the logic prescribed by the preceding Sub-Function (i.e. 2-Way or No-Match). Upon completion, the Program Module executes the Record Estimated Accrual Item Block Sub-Function.

Record Estimated Accrual Item Block. The Program Module creates the Estimated Accrual Item Block after executing the Record Estimated Accrual Control Block Sub-Function.

Record Estimated Accrual Account Block. The Program Module creates the Estimated Accrual Account Block after executing the Record Estimated Accrual Item Block Sub-Function.

Record General Ledger. The Program Module creates the CFS General Ledger entries for each Purchase Order Account Block created in the previous Sub-Function.

Is There Another Line Item? If an unconverted line item exists, the Program Module executes the Record Estimated Accrual Item Block Sub-Function. If no unconverted line item exists (i.e. all line items have been converted), the Program Module executes the Record Invoice Shell Function.

Record Invoice Shell. When applicable line items have been converted, the Is There Another Line Item Sub-Function executes the Record Invoice Shell Function.

6.3 Record Estimated Accrual Function Processing

6.3.1 Record Estimated Accrual Control Block Function

The Program Module will only need to populate **EA CONTROL** once for each converted document.

The following section explains how to obtain the values needed to populate the Estimated Accrual Control Block. Because the derived values differ based on the document matching criteria, this Sub-Function is broken into No-Match **EA CONTROL** and 2-Way match **EA CONTROL**.

6.3.1.1 Logic

Table 6 provides a field by field description of how to obtain the values to populate **EA CONTROL**. The Field Name column contains every field name present on **EA CONTROL**. The Source For No-Match **EA CONTROL** column contains information for determining how the corresponding field name is populated when the selected document is identified as a No-Match document.

The Source For 2-Way **EA CONTROL** contains information for determining how the corresponding field name is populated when the selected document is identified as a 2-Way match document. 2-Way match Estimated Accruals use the Purchase Order information with the greatest **Change order Number** when obtaining data from the Purchase Order tables.

Table 6. EA CONTROL		
Field Name	Source For No-match EA CONTROL	Source For 2-Way EA CONTROL
Ea_control_id	select EA_CONTROL_ID_SEQ.NEXTVAL from DUAL	select EA_CONTROL_ID_SEQ.NEXTVAL from DUAL
Document_source	NONE	OBLIG
Gl_end_date	09/30/2002	09/30/2002
Bureau_code	CONVERT_ACTDOC_ACCOUNT. Bureau_code	select Bureau_code from PO_CONTROL PO, where PO.Document_no equals current Document_no and PO.Change_order_no is the max Change_order_no
All_accrued_flag	N	N
Ap_status	O	O
Country_code	select Country_code (see source for Contact in section 5.3.2.1)	PO_CONTROL.Country_code
Discount_amount1	select Discount_amount1 (see source for Contact in section 5.3.2.1)	PO_CONTROL.Discount_amount1
Discount_amount2	select Discount_amount2 (see source for Contact in section 5.3.2.1)	PO_CONTROL.Discount_amount2
Discount_days1	select Discount_days1 (see source for Contact in section 5.3.2.1)	PO_CONTROL.Discount_days1

Table 6. EA CONTROL		
Field Name	Source For No-match EA CONTROL	Source For 2-Way EA CONTROL
Discount_days2	select Discount_days2 (see source for Contact in section 5.3.2.1)	PO_CONTROL.Discount_days2
Discount_flag1	select Discount_flag1 (see source for Contact in section 5.3.2.1)	PO_CONTROL.Discount_flag1
Discount_flag2	select Discount_flag2 (see source for Contact in section 5.3.2.1)	PO_CONTROL.Discount_flag2
Document_status	OPEN	OPEN
Exchange_rate	Null	Null
Net_days1	select Net_days1 (see source for Contact in section 5.3.2.1)	PO_CONTROL.Net_days1
Net_days2	select Net_days2 (see source for Contact in section 5.3.2.1)	PO_CONTROL.Net_days2
Notes	"This document has been converted from FIMA. To view the document in FIMA, please use the CFS Source Reference Number as the FIMA Document Number."	"This document has been converted from FIMA. To view the document in FIMA, please use the CFS Source Reference Number as the FIMA Document Number."
Po_date	Null	09/30/2002
Po_no	0	PO_CONTROL.Document_no
Po_type	NONE	PO_CONTROL.Document_type

Table 6. EA CONTROL		
Field Name	Source For No-match EA CONTROL	Source For 2-Way EA CONTROL
Reference_no	select Document_no from CONVERT_ACTDOC	PO_CONTROL.Reference_no
Release_no	0	PO_CONTROL.Release_no
Status_date	09/30/2002	09/30/2002
Trans_date	09/30/2002	09/30/2002
Vendor_id	CONVERT_ACTDOC. Vendor_id	PO_CONTROL.Vendor_id
Vendor_name	select Vendor_name (see source for Contact in section 5.3.2.1)	select Vendor_name (see source for Contact in section 5.3.2.1)
Vendor_no	CONVERT_ACTDOC. Vendor_no	PO_CONTROL.Vendor_no
Approved_flag	Y	Y
Approved_by	select First_Name, Last_Name from EMPLOYEE_CONTROL where Emp_no equals current Emp_no	PO_CONTROL.Placed_by
Approved_emp_no	current Emp_no	PO_CONTROL.Placed_emp_no
Approved_date	09/30/2002	09/30/2002
Reverse_ea_flag	Null	Null
Reverse_ea_control_id	Null	Null
User_name	CONVERSION	CONVERSION
Modification_date	sysdate	sysdate

Table 6. EA CONTROL		
Field Name	Source For No-match EA CONTROL	Source For 2-Way EA CONTROL
Device_name	CONVERSION	CONVERSION
Invoice_type	Null	CVINV
Invoice_no	Null	Null

6.3.1.2 Output

Upon the first completion of the Estimated Accrual Control Block Sub-Function, the Program Module:

- Inserts to **EA CONTROL**.
- Sets the **CFS Ea Control Id** field on **CONVERT ACTDOC** equal to the **Ea Control Id** (Ea_control_id) value from the recently created **EA CONTROL** record.

6.3.1.3 Error Handling

If an error is encountered while inserting into **EA CONTROL**, write an 'Unable to update **EA CONTROL** Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'. In addition, the Program Module aborts converting the current document and removes previously inserted data for the current document.

6.3.2 Record Estimated Accrual Item Block

The Program Module records an entry to **EA ITEM** for records with an **Unpaid Accrued Expenditure** value greater than zero on **CONVERT ACTDOC ACCOUNT**. The relationship between the **EA ITEM** table and the **EA CONTROL** table is the **EA Control Id**.

The following Sub-Functions detail how to populate the Estimated Accrual Item table. Because the derived values differ based on the document matching criteria, this Sub-Function is broken into No-Match **EA ITEM** and 2-Way match **EA ITEM**.

6.3.2.1 Logic

Table 7 provides a field by field description of how to obtain the values to populate **EA ITEM**. The Field Name column contains the fields present on **EA ITEM**. The Source For No-Match **EA ITEM** column contains information for determining how the corresponding field name is populated when the selected document is identified as a No-Match document.

The Source For 2-Way Match **EA ITEM** column contains information for determining how the corresponding field name is populated when the selected document is identified as a 2-Way match document. 2-Way match Estimated Accruals use the Purchase Order information with the greatest **Change order Number** when obtaining data from the Purchase Order tables. Only line items on a 2-Way match document with an **Unpaid Accrued Expenditure** value greater than zero on **CONVERT ACTDOC ACCOUNT** are converted to **EA ITEM**.

Table 7. EA ITEM

Field Name	Source For No-Match EA ITEM	Source For 2-Way Match EA ITEM
Ea_item_id	select Ea_item_id_seq.NEXTVAL from DUAL	select Ea_item_id_seq.NEXTVAL from DUAL
Ea_control_id	EA_CONTROL. Ea_control_id	EA_CONTROL. Ea_control_id
Item_no	This value begins with one and is incremented by one for each EA ITEM entry. Initialized to one for each unique Ea_control_id.	CONVERT_ACTDOC_ACCOUNT.cfs_item_no
Unit_price	CONVERT_ACTDOC_ACCOUNT.Unpaid_accrued_exp	CONVERT_ACTDOC_ACCOUNT.Unpaid_accrued_exp
Payment_office_code	Null	Null

Table 7. EA ITEM		
Field Name	Source For No-Match EA ITEM	Source For 2-Way Match EA ITEM
Commodity_code	current Commodity_code	select Commodity_code from PO_ITEM PI, CONVERT_ACTDOC CA where PI.Document_no equals CA.Cfs_document_number and PI.Item_no equals CA.Item_no and PI.Change_order_no equals max Change_order_no
Ap_status	O	O
Emp_no	Null	Null
Item_type	current Item_type	select Item_type (see Commodity_code)
Qty_accrued	1	1
Qty_accrued_todate	Null	Null
Amount_accrued	CONVERT_ACTDOC_ACCOUNT.Unpaid_accrued_exp	CONVERT_ACTDOC_ACCOUNT.Unpaid_accrued_exp
Amount_accrued_todate	Null	0
Ordered_qty	Null	1
Over_tolerance	N	N
Holdback_amount	Null	Null
Holdback_type	Null	Null
Over_tolerance_qty	Null	Null

Table 7. EA ITEM		
Field Name	Source For No-Match EA ITEM	Source For 2-Way Match EA ITEM
Over_tolerance_amount	Null	Null
Ri_method	D	D
User_name	CONVERSION	CONVERSION
Modification_date	sysdate	sysdate
Device_name	CONVERSION	CONVERSION

6.3.2.2 *Output*

Upon completion of the Estimated Accrual Item Block Sub-Function, the Program Module:

- Inserts to **EA ITEM**.
- Sets the **CFS Ea Item Id** field on **CONVERT ACTDOC ACCOUNT** equal to the **Ea Item Id** (Ea_item_id) value from the recently created **EA ITEM** record.
- Updates the **Accrued Amount** (**Accrued_amount**) field on **PO ACCOUNT** for the current **Item Number**. The **Accrued Amount** field equals the value in the **Amount Accrued** field on **EA ITEM**. The update need only occur on the **PO ACCOUNT** record with the greatest **Change Order Number** for the given **Item Number**.
- Updates the **Accrued Quantity** (**Accrued_qty**) field on **PO ACCOUNT** for the current Item Number. The Accrued Quantity field equals the value in the **Quantity Accrued** (Qty_accrued) field on **EA ITEM**. The update need only occur on the **PO ACCOUNT** record with the greatest **Change Order Number** for the given **Item Number**.

6.3.2.3 *Error Handling*

If an error is encountered while inserting into **EA ITEM**, write an 'Unable to update **EA ITEM** Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'. In addition, the Program Module aborts converting the current document and removes previously inserted data for the current document.

6.3.3 *Record Estimated Accrual Account Block*

The Program Module records an individual entry to **EA ACCOUNT** for each individual entry to **EA ITEM**. The relationship between the **EA ACCOUNT** table and the **EA ITEM** table is the **EA Item Id**.

The following Sub-Functions detail how to populate the Estimated Accrual Account table. Because the derived values differ based on the document matching criteria, this Sub-Function is broken into No-Match EA Account and 2-Way match EA Account.

6.3.3.1 *Logic*

Table 8 provides a field by field description of how to obtain the values to populate **EA ACCOUNT**. The Field Name column contains the fields present on **EA ACCOUNT**. The Source For No-Match **EA ACCOUNT** column contains information for determining how the corresponding field name is populated when the selected document is identified as a No-Match document.

The Source For 2-Way Match **EA ACCOUNT** column contains information for determining how the corresponding field name is populated when the selected document is identified as a 2-Way match document. 2-Way match Estimated Accruals use the Purchase Order information with the greatest **Change order Number** when obtaining data from the Purchase Order tables.

Table 8. EA ACCOUNT

Field Name	Source For No-Match EA ACCOUNT	Source For 2-Way Match EA ACCOUNT
Ea_account_id	Select Ea_account_id_seq.NEXTVAL from DUAL	Select Ea_account_id_seq.NEXTVAL from DUAL
Ea_item_id	EA_ITEM.Ea_item_id	EA_ITEM.Ea_item_id
Ea_control_id	EA_ITEM.Ea_control_id	EA_ITEM.Ea_control_id
Item_no	current Item_no	current Item_no
Line_no	1	1

Table 8. EA ACCOUNT		
Field Name	Source For No-Match EA ACCOUNT	Source For 2-Way Match EA ACCOUNT
Fiscal_year	CONVERT_ACTDOC_ACCOUNT.Fiscal_year	CONVERT_ACTDOC_ACCOUNT.Fiscal_year
Display_fiscal_year	Last two digits of CONVERT_ACTDOC_ACCOUNT.Fiscal_year	Last two digits of CONVERT_ACTDOC_ACCOUNT.Fiscal_year
Amount	CONVERT_ACTDOC_ACCOUNT.Unpaid_accrued_exp	CONVERT_ACTDOC_ACCOUNT.Unpaid_accrued_exp
Ap_amount	Null	Null
Ap_qty	Null	Null
Ap_status	O	O
Direct_flag	select Direct_flag from PROJECT where PROJECT.Project_code equals CONVERT_ACTDOC_ACCOUNT.Project_code	PO_ACCOUNT.Direct_flag
Final_flag	Null	N
Percent	100	100
Qty_ordered	Null	1
Qty_accrued	1	1
Qty_remaining	Null	1
Amount_ordered	Null	Null
Amount_accrued	Null	Null
Amount_remaining	Null	Null
Trans_code	Null	Null

Table 8. EA ACCOUNT		
Field Name	Source For No-Match EA ACCOUNT	Source For 2-Way Match EA ACCOUNT
Bureau_code	CONVERT_ACTDOC_ACCOUNT.Bureau_code	PO_ACCOUNT.Bureau_code
Fund_code	CONVERT_ACTDOC_ACCOUNT.Fund_code	PO_ACCOUNT.Fund_code
Org1_code	CONVERT_ACTDOC_ACCOUNT.Org1_code	PO_ACCOUNT.Org1_code
Org2_code	CONVERT_ACTDOC_ACCOUNT.Org2_code	PO_ACCOUNT.Org2_code
Org3_code	CONVERT_ACTDOC_ACCOUNT.Org3_code	PO_ACCOUNT.Org3_code
Org4_code	CONVERT_ACTDOC_ACCOUNT.Org4_code	PO_ACCOUNT.Org4_code
Org5_code	CONVERT_ACTDOC_ACCOUNT.Org5_code	PO_ACCOUNT.Org5_code
Org6_code	CONVERT_ACTDOC_ACCOUNT.Org6_code	PO_ACCOUNT.Org6_code
Org7_code	CONVERT_ACTDOC_ACCOUNT.Org7_code	PO_ACCOUNT.Org7_code
Program1_code	CONVERT_ACTDOC_ACCOUNT.Program1_code	PO_ACCOUNT. Program1_code
Program2_code	CONVERT_ACTDOC_ACCOUNT.Program2_code	PO_ACCOUNT. Program2_code
Program3_code	CONVERT_ACTDOC_ACCOUNT.Program3_code	PO_ACCOUNT. Program3_code
Program4_code	CONVERT_ACTDOC_ACCOUNT.Program4_code	PO_ACCOUNT. Program4_code

Table 8. EA ACCOUNT

Field Name	Source For No-Match EA ACCOUNT	Source For 2-Way Match EA ACCOUNT
Project_code	CONVERT_ACTDOC_ACCOUNT.Project_code	PO_ACCOUNT.Project_code
Task_code	CONVERT_ACTDOC_ACCOUNT.Task_code	PO_ACCOUNT.Task_code
Object1_code	CONVERT_ACTDOC_ACCOUNT Object1_code	PO_ACCOUNT. Object1_code
Object2_code	CONVERT_ACTDOC_ACCOUNT Object2_code	PO_ACCOUNT. Object2_code
Object3_code	CONVERT_ACTDOC_ACCOUNT Object3_code	PO_ACCOUNT. Object3_code
Object4_code	CONVERT_ACTDOC_ACCOUNT Object4_code	PO_ACCOUNT. Object4_code
User_define_accs	0	0
Approved_by	Null	Null
Approved_date	Null	Null
Approved_emp_no	Null	Null
Approved_flag	Null	Null
Approved_title	Null	Null
Approved_work_phone_no	Null	Null
Override_by	Null	Null
Override_date	Null	Null
Override_emp_no	Null	Null
Override_flag	Null	Null
Override_title	Null	Null

Table 8. EA ACCOUNT		
Field Name	Source For No-Match EA ACCOUNT	Source For 2-Way Match EA ACCOUNT
Override_work_phone_no	Null	Null
User_name	CONVERSION	CONVERSION
Modification_date	sysdate	sysdate
Device_name	CONVERSION	CONVERSION

6.3.3.2 *Output*

Upon the completion of the Estimated Accrual Account Block Sub-Function, the Program Module:

- Inserts to **EA ACCOUNT**.
- Sets the **CFS Ea Account Id** field on **CONVERT ACTDOC ACCOUNT** equal to the **Ea Account Id** (Ea_account_id) value from the recently created **EA ACCOUNT** record.

6.3.3.3 *Error Handling*

If an error is encountered while inserting into **EA ACCOUNT**, write an 'Unable to update **EA ACCOUNT** Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'. In addition, the Program Module aborts converting the current document and removes previously inserted data for the current document.

6.3.4 *Record General Ledger*

The Program Module records an entry to the CFS General Ledger after completion of the Record Estimated Accrual Account Block Function. The following Sub-Functions detail how to populate the CFS General Ledger table.

6.3.4.1 *Logic*

Table 9 provides a field by field description of how to obtain the values to populate **TRIAL**. The Field Name column contains the fields present on **TRIAL**. The Source For EA **TRIAL** column contains information for determining how the corresponding field name is populated.

When a TRIAL entry is created, several fields are derived differently based of the document matching criteria. For this reason Table 9 includes terminology to help identify the specific field values that are derived differently based on the document matching criteria. Beneath the Source For EA **TRIAL** column, field values that must be derived for No-Match documents are designated with the term '*No-Match:*' followed by the code necessary to acquire the modified value. Field values that must be derived for 2-Way match documents are designated with the term '*2-Way:*' followed by the code necessary to acquire the modified value.

For example, the field value **Original Document Number** (Org_Document_No) is equal to the **EA Control Id** when the current document is a No-Match document. Whereas the same field value, **Original Document Number**, is equal to the CFS **Document Number** when the current document is a 2-Way match document.

If no special designation is present, the field value is derived the same for No-Match and 2-Way match documents.

Table 9. TRIAL

Field Name	Source For EA TRIAL Table
Gl_trans_no	Select TRIAL_SEQ.NEXTVAL from DUAL
Trans_date	09/30/2002
Gl_end_date	09/30/2002
Fiscal_quarter	4
Fiscal_year	2002
Fy_end_date	09/30/2002
Subsystem_code	EA
Trans_source	EA
Trans_ref	'EA-'||lpad(to_char(EA_CONTROL.EA_CONTROL_ID), 8, '0')

<table>
<tr><th colspan="2">Table 9. TRIAL</th></tr>
<tr><th>Field Name</th><th>Source For EA TRIAL Table</th></tr>
<tr><td>Trans_descr</td><td>'TC:'||TC.trans_code||' - '||’OPEN DOCUMENT CONVERSION’ ||' '|TC.trans_descr
from TRANS_CONTROL TC</td></tr>
<tr><td>Trans_no</td><td>EA_CONTROL.Ea_control_id</td></tr>
<tr><td>Trans_type_flag</td><td>G</td></tr>
<tr><td>Direct_flag</td><td>EA_ACCOUNT.Direct_flag</td></tr>
<tr><td>Stat_unit_code</td><td>Null</td></tr>
<tr><td>Stat_unit_qty</td><td>Null</td></tr>
<tr><td>Emp_no</td><td>Null</td></tr>
<tr><td>Hourly_rate</td><td>Null</td></tr>
<tr><td>Debit_amount</td><td>To Be Determined</td></tr>
<tr><td>Credit_amount</td><td>To Be Determined</td></tr>
<tr><td>Budget_amount</td><td>To Be Determined</td></tr>
<tr><td>Fund_code_fiscal_year</td><td>EA_ACCOUNT.Fiscal_year</td></tr>
<tr><td>Fund_code</td><td>EA_ACCOUNT.Fund_code</td></tr>
<tr><td>Bureau_code</td><td>EA_ACCOUNT.Bureau_code</td></tr>
<tr><td>Org1_code</td><td>EA_ACCOUNT.Org1_code</td></tr>
<tr><td>Org2_code</td><td>EA_ACCOUNT.Org2_code</td></tr>
<tr><td>Org3_code</td><td>EA_ACCOUNT.Org3_code</td></tr>
<tr><td>Org4_code</td><td>EA_ACCOUNT.Org4_code</td></tr>
<tr><td>Org5_code</td><td>EA_ACCOUNT.Org5_code</td></tr>
<tr><td>Org6_code</td><td>EA_ACCOUNT.Org6_code</td></tr>
<tr><td>Org7_code</td><td>EA_ACCOUNT.Org7_code</td></tr>
</table>

Table 9. TRIAL	
Field Name	Source For EA TRIAL Table
Program1_code	EA_ACCOUNT.Program1_code
Program2_code	EA_ACCOUNT.Program2_code
Program3_code	EA_ACCOUNT.Program3_code
Program4_code	EA_ACCOUNT.Program4_code
Project_code	EA_ACCOUNT.Project_code
Task_code	EA_ACCOUNT.Task_code
Object1_code	EA_ACCOUNT.Object1_code
Object2_code	EA_ACCOUNT.Object2_code
Object3_code	EA_ACCOUNT.Object3_code
Object4_code	EA_ACCOUNT.Object4_code
User_define_accs	0
Account_no	To Be Determined
Sub_account_no	To Be Determined

Table 9. TRIAL

Field Name	Source For EA TRIAL Table
Full_account_no	EA_ACCOUNT.Bureau_code, EA_ACCOUNT.Project, ‘-‘, EA_ACCOUNT.Task_code, EA_ACCOUNT.Fund_Code, EA_ACCOUNT.Program1_Code, ‘-‘, EA_ACCOUNT.Program2_Code, ‘-‘, EA_ACCOUNT.Program3_Code, ‘-‘, EA_ACCOUNT.Program4_Code, EA_ACCOUNT.Org1_code, EA_ACCOUNT.Org2_code, EA_ACCOUNT.Org3_code, EA_ACCOUNT.Org4_code, EA_ACCOUNT.Org5_code, EA_ACCOUNT.Org6_code, EA_ACCOUNT.Org7_code, EA_ACCOUNT.Object1_Code, ‘-‘, EA_ACCOUNT.Object2_Code, ‘-‘, EA_ACCOUNT.Object3_Code, ‘-‘, EA_ACCOUNT.Object4_Code, ‘000000', EA_ACCOUNT.Account_no, ‘-‘, EA_ACCOUNT. Sub_Account_No
Apport_category	A
Apportionment_no	Null
Advice_no	Null
Avail_begin_date	Null
Avail_end_date	Null
Category_item_no	0
Begbal_flag	N
Document_type	EA
Document_no	EA_CONTROL.Ea_control_id
Document_date	09/30/2002

Table 9. TRIAL	
Field Name	Source For EA TRIAL Table
Change_order_no	0
Release_no	EA_CONTROL.Release_no
Vendor_no	EA_CONTROL.Vendor_no
Vendor_id	EA_CONTROL.Vendor_id
Customer_no	Null
Contact_no	Null
Reference_no	EA_CONTROL.Reference_no
Item_type	EA_ITEM.Item_type
Item_no	EA_ITEM.Item_no
Line_no	1
Original Fields Will Be Formatted If the Account_no equals ‘4901'	
Org_document_date	09/30/2002
Org_document_no	*No-Match:* EA_CONTROL.Ea_control_id *2-Way:* When inserting an entry for Account_no 4801 this value is CONVERT_ACTDOC.Cfs_document_no When inserting an entry for Account_no 4901 this value is EA_CONTROL.Ea_control_id
Org_release_no	EA_CONTROL.Release_no
Org_item_type	EA_ITEM.Item_type
Org_item_no	EA_ITEM.Item_no
Org_line_no	1
User_name	CONVERSION

Table 9. TRIAL	
Field Name	Source For EA TRIAL Table
Modification_date	sysdate
Device_name	CONVERSION
Financing_flag	Null
SF224_required	Null
Agency_location_code	Null
Cash_trans_type	Null
Accomplished_date	Null
Trial_id	Null (Updated via a trigger on trial)
Trans_code	To be determined
Reorg_token	Null
Account_no_digit_1	first digit of Account_no from TRIAL
Account_no_digit_2	second digit of Account_no from TRIAL

6.3.4.2 *Output*

Upon the completion of the Record General Ledger Sub-Function, the Program Module:

- Inserts to TRIAL.

6.3.4.3 *Error Handling*

If an error is encountered while inserting into **TRIAL**, write an 'Unable to update TRIAL Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'. In addition, the Program Module aborts converting the current document and removes previously inserted data for the current document.

6.3.5 *Is There Another Line Item?*

If an unconverted line item exists, the Program Module executes the Record Estimated Accrual Item Block Sub-Function. If no unconverted line items exist (i.e. all line items have been converted), the Program Module executes the Record Invoice Shell Function.

6.3.5.1 Logic

For No-Match documents, the Program Module increments the cursor to the next record from **CONVERT ACTDOC ACCOUNT**. If a record exists, the Program Module executes the Record Estimated Accrual Item Sub-Function.

If no record exists (i.e. the Program Module has processed all the line items in **CONVERT ACTDOC ACCOUNT**), the Program Module executes the Record Invoice Shell Function.

For 2-Way match documents, the Program Module increments the cursor to the next record from **CONVERT ACTDOC ACCOUNT**. When a record exists, the Program Module determines whether the line item has an Unpaid Accrued Expenditure value greater than zero. If the Unpaid Accrued Expenditure value is greater than zero, the Program Module executes the Record Estimated Accrual Item Sub-Function.

If the Unpaid Accrued Expenditure value is not greater than zero, the Program Module increments the cursor to the next record from **CONVERT ACTDOC ACCOUNT** and re-executes the Is There Another Line Item Sub-Function. This process is continued until no records exist (i.e. the Program Module has processed applicable line items in **CONVERT ACTDOC ACCOUNT**) and the Program Module executes the Record Invoice Shell Function.

7 Record Invoice Shell Function

Upon completion of the Record Estimated Accrual Function, the Program Module executes the Record Invoice Shell Function.

7.1 Purpose of Record Invoice Shell Function

The purpose of the Record Invoice Shell Function is to replicate CFS processing that occurs upon approval of an Estimated Accrual transaction. When a user approves an Estimated Accrual, the system populates select fields on the invoice tables (**AP CONTROL**, **AP DETAIL**), with information from the Estimated Accrual transaction. The Record Invoice Shell Function details which invoice fields are populated and how the data is derived. This Function adds line items to the Invoice Shell that were previously converted during the Record Estimated Accrual Function.

7.2 Process Flow of Invoice Shell Order Function

This section describes an overview of the major steps needed to insert an Estimated Accrual Invoice Shell in to CFS. The process is illustrated in Figure 5.

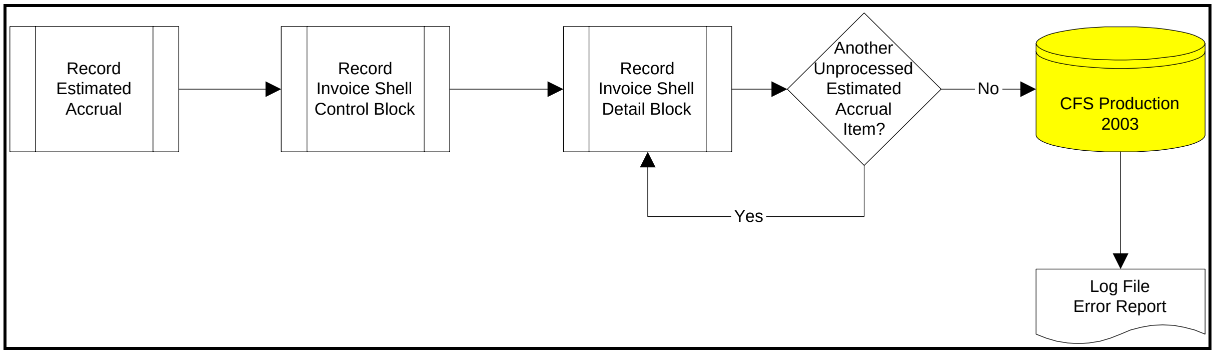

Figure 5. Record Invoice Shell Process Flow

Record Estimated Accrual. After the successful completion of the Record Estimated Accrual Function, the Program Module executes the Record Invoice Shell Function.

Record Invoice Shell Control Block. The Program Module creates a single entry on the Invoice Control Block (**AP CONTROL**) that derives data from the previously converted Estimated Accrual Control Block (**EA CONTROL**).

Record Invoice Shell Detail Block. With the Invoice Shell Control Block created, the Program Module creates an entry on the Invoice Detail Block **(AP DETAIL**) for an entry on the Estimated Accrual Item Block.

Another Unprocessed Estimated Accrual Item? The Program Module must access whether there is another Item Number from the Estimated Accrual Item Block that must also be entered on the Invoice Shell Detail Block. If another Item Number exists that is not on the Invoice Shell Detail Block, the Record Invoice Shell Detail Block Sub-Function is executed.

When all Item Numbers from the Estimated Accrual Item Block appear on the Invoice Shell Detail Block, the document is inserted into CFS Production 2003 and executes the Crosswalk Document Data Function.

7.3 Business Rules of Record Invoice Shell Function

The following business rules pertain to this Program Module:

- The system posts invoice shell records as unapproved.
- The invoice shell is populated with a purchase address for a given vendor. The user will update vendor to reflect a payment address prior to approval.

7.4 Input to Record Invoice Shell Function

The following tables are input to the Record Invoice Shell Function:

- **EA CONTROL**
- **EA ITEM**
- **EA ACCOUNT**

7.5 Record Invoice Shell Function Processing

The Program Module will only need to populate **AP CONTROL** once, deriving the necessary data from the recently created **EA CONTROL** table. The Program Module populates **AP DETAIL** once for each **EA ITEM** record created. Data for **AP DETAIL** will be derived from the recently created **EA ITEM** and **EA ACCOUNT** records. The Record Invoice Shell Control Block Function will only be called if the Program Module created an Estimated Accrual record.

7.5.1 Record Invoice Shell Control Block

7.5.1.1 Logic

Table 10 provides a field by field description of how to obtain the values to populate **AP CONTROL**. The Field Name column contains every field name present on **AP CONTROL**. The Source For AP Control column contains information for determining how the corresponding field name is populated.

When an **AP CONTROL** entry is created, several fields are derived differently based of the document matching criteria. For this reason, Table 10 includes terminology to help identify the specific field values that are derived differently based on the document matching criteria. Beneath the Source For **AP CONTROL** column, field values that must be derived for No-Match documents are designated with the term '*No-Match:*' followed by the code necessary to acquire the modified value.

Field values that must be derived for 2-Way match documents are designated with the term '*2-Way:*' followed by the code necessary to acquire the modified value.

If no special designation is present, the field value is derived the same for No-Match and 2-Way match documents.

Table 10. AP CONTROL

Field Name	Source For AP CONTROL
Trans_no	select Sequence_no + 1 from MAXSEQNOS where table_name equals ‘ AP_CONTROL’
Invoice_no	SHELL EA:0000[EA_CONTROL.Ea_control_id]. If the Ea_control_id is more than 4 digits, drop zeros from after the ':' as needed.
Sub_invoice_no	0
Vendor_no	EA_CONTROL.Vendor_no
Vendor_id	EA_CONTROL.Vendor_id
Vendor_name	Null
Address1	Null
Address2	Null
Address_state	Null
Bureau_code	EA_CONTROL.Bureau_code
Approp_symbol	Null
Agency_Location_code	Null
City	Null
Created_by	select Db_user_name from EMPLOYEE_CONTROL where Emp_no equals EA_CONTROL.Approved_emp_no
Created_date	09/30/2002

Table 10. AP CONTROL	
Field Name	Source For AP CONTROL
Country_code	EA_CONTROL.Country_code
Discount_flag1	EA_CONTROL.Discount_flag1
Discount_days1	EA_CONTROL.Discount_days1
Discount_amount1	EA_CONTROL.Discount_amount1
Discount_flag2	EA_CONTROL.Discount_flag2
Discount_days2	EA_CONTROL.Discount_days2
Discount_amount2	EA_CONTROL.Discount_amount2
Fastpay_flag	N
Hold_disc_term_used	Null
Invoice_date	Null
Invoice_status	OPEN
Invoice_type	CVINV
Net_days1	EA_CONTROL.Net_days1
Net_days2	EA_CONTROL.Net_days2
Net_invoice_amount	Sum EA_ITEM.Amount_accrued for all EA ITEM records
Document_source	*No-Match:* NONE *2-Way:* OBLIG
Po_no	*No-Match:* 0 *2-Way:* PO_CONTROL.Document_no
Po_type	*No-Match:* NONE *2-Way:* PO_CONTROL.Document_type

Table 10. AP CONTROL	
Field Name	Source For AP CONTROL
Prompt_pay_flag	select Prompt_pay_flag from VENDOR_DETAIL VD where VD.Vendor_no equals current Vendor_no and VD.Vendor_id equals current Vendor_id
Received_date	Null
Reference_no	EA_CONTROL.Reference_no
Release_no	Null
Returned_date	Null
Status_date	09/30/2002
Zip_code	Null
Approved_flag	N
Approved_by	Null
Approved_emp_no	Null
Approved_date	Null
User_name	CONVERSION
Modification_date	sysdate
Device_name	CONVERSION
Invoice_amount	Null
Document_type	Null
Document_no	Null
Passed_screen_edit	Null
Po_change_order_no	Null
Pop_from_date	Null
Pop_to_date	Null

Table 10. AP CONTROL	
Field Name	Source For AP CONTROL
Pop_active_flag	Null

7.5.1.2 Error Handling

If an error is encountered while inserting into **AP CONTROL**, write an 'Unable to update **AP CONTROL** Error' to the **ERROR CONVERSION** table and update the Error Flag to 'Y'. In addition, the Program Module aborts converting the current document and removes previously inserted data for the current document.

7.5.2 Record Invoice Shell Detail Block

7.5.2.1 Logic

Table 11 provides a field by field description of how to obtain the values to populate **AP DETAIL**. The Field Name column contains every field name present on **AP DETAIL**. The Source For **AP DETAIL** column contains information for determining how the corresponding field name is populated.

Table 11. AP DETAIL	
Field Name	Source For AP DETAIL
Trans_no	AP_CONTROL.Trans_no
Item_no	EA_ITEM.Item_no
Item_type	EA_ITEM.Item_type
Line_no	1
Accrual_amount	Null
Accrual_qty	Null
Accepted_date	Null
Ac_flag	N
Adv_trans_no	Null
Adv_account_no	Null

Table 11. AP DETAIL	
Field Name	Source For AP DETAIL
Adv_sub_account_no	Null
Adv_invoice_type	Null
Adv_invoice_no	Null
Adv_invoice_sub_no	Null
Amount	EA_ITEM.Amount_accrued
Bank_code	TREAS
Center_code	Null
Default_matching_flag	Y
Direct_flag	EA_ACCOUNT.Direct_flag
Discount_date	Null
Discount_flag	N
Discount_lost	Null
Display_fiscal_year	EA_ACCOUNT.Display_fiscal_year
Dl_code	Null
Document_type	Null
Document_no	Null
Due_date	Null
Emp_no	0
Exchange_rate	Null
Fco_emp_no	Null
Final_flag	N
Fiscal_year	EA_ACCOUNT.Fiscal_year
Gl_end_date	09/30/2002

Table 11. AP DETAIL	
Field Name	Source For AP DETAIL
Gl_impact_source	PM050
Holdback_amount	Null
Holdback_flag	N
Holdback_type	Null
Hold_disc_amt1	Null
Hold_disc_amt2	Null
Hold_disc_date1	Null
Hold_disc_date2	Null
Hold_take_disc1	Null
Hold_take_disc2	Null
Item_descr	Null
Lr_code	Null
Line_status	OPEN
Orig_flag	Y
Over_tolerance_flag	N
Payment_enclosure	Null
Payment_office_code	Null
Payment_process_flag	N
Payment_process_date	Null
Po_amount	Null
Po_ap_amount	Null
Po_flag	N
Prompt_pay_type	current Prompt_pay_type

Table 11. AP DETAIL	
Field Name	Source For AP DETAIL
Qty	1
Recorded_in_gl_flag	Y
Release_flag	N
Rt_flag	N
Rt_no	Null
Schedule_date	Null
Schedule_no	0
Schedule_type	MISC
Status_date	09/30/2002
Trans_code	NONE
Unit_price	EA_ITEM.Unit_price
V1099_flag	select W9_received from VENDOR_DETAIL VD where VD.Vendor_no equals current Vendor_no and Vendor_id equals current Vendor_id
Withhold_flag	Null
Withhold_percent	Null
Bureau_code	EA_ACCOUNT.Bureau_code
Fund_code	EA_ACCOUNT.Fund_code
Org1_code	EA_ACCOUNT.Org1_code
Org2_code	EA_ACCOUNT.Org2_code
Org3_code	EA_ACCOUNT.Org3_code
Org4_code	EA_ACCOUNT.Org4_code
Org5_code	EA_ACCOUNT.Org5_code

Table 11. AP DETAIL	
Field Name	Source For AP DETAIL
Org6_code	EA_ACCOUNT.Org6_code
Org7_code	EA_ACCOUNT.Org7_code
Program1_code	EA_ACCOUNT.Program1_code
Program2_code	EA_ACCOUNT.Program2_code
Program3_code	EA_ACCOUNT.Program3_code
Program4_code	EA_ACCOUNT.Program4_code
Project_code	EA_ACCOUNT.Project_code
Task_code	EA_ACCOUNT.Task_code
Object1_code	EA_ACCOUNT.Object1_code
Object2_code	EA_ACCOUNT.Object2_code
Object3_code	EA_ACCOUNT.Object3_code
Object4_code	EA_ACCOUNT.Object4_code
User_define_accs	0
Discount_calc_flag	Null
Wh1099_calc_flag	Null
Holdback_calc_flag	Null
Interest_calc_flag	Null
Interest_date	Null
Reissue_flag	Null
User_name	CONVERSION
Modification_date	sysdate
Device_name	CONVERSION
Rt_no_for_reverse	Null

Table 11. AP DETAIL	
Field Name	**Source For AP DETAIL**
Ac_inv_flag	Null
Ri_method	D
Po_ap_qty	Null
Interest_rate	Null
Interest_date	Null
Sub_trans_no	Null
Apc_flag	Null
Ea_control_id	select Cfs_ea_control_id from CONVERT_ACTDOC.
Ea_control_id_for_reverse	select Cfs_ea_control_id from CONVERT_ACTDOC.
Po_accrual_amount	Null
Po_accrual_qty	Null
Shell_adj_amount	Null

7.5.2.2 Error Handling

If an error is encountered while inserting into **AP DETAIL**, write an 'Unable to update **AP DETAIL** Error' to the **ERROR CONVERSION** table and update the **Error Flag** to 'Y'. In addition, the Program Module aborts converting the current document and removes previously inserted data for the current document.

7.5.3 Is There Another Unprocessed Estimated Accrual Line Item

7.5.3.1 Logic

The Program Module verifies that all Item Numbers from **EA ITEM** have been inserted in **AP DETAIL**. If this condition is not met, the Program Module loops back to the Record Invoice Shell Detail Function and insert the appropriate data for the unprocessed Item Number.

If all Item Numbers from **EA ITEM** have been inserted in **AP DETAIL**, the Program Module:

- Updates the **Conversion Status Flag** on **CONVERT ACTDOC** to 'Y'
- Sets the **Conversion Date** equal to the system date
- Executes the Crosswalk Document Data Function to determine the next **FIMA Document Number** eligible for conversion.

8 Core Financial System (CFS) Set-Up

The following data must be established in CFS prior to executing the conversion Program Module:

8.1 Document Type

Each **CFS Document Type** corresponding to a 2-Way match or No-Match document is created in CFS. The required information is displayed in Appendix A. The appropriate **CFS Document Type** must be entered on the *ACCOUNTING SYSTEM CODE MAINTENANCE SCREEN (GL021),* under the accounting system code 'OBLIG'.

8.2 Item Type

Each **CFS Item Type** identified in Appendix B, must be entered on the *ACCOUNTING SYSTEM CODE MAINTENANCE SCREEN (GL021)* under the accounting system code 'CDITEM'. In addition,

8.3 Prompt Pay Type

The required **Prompt Pay Type(s)** are currently established in CFS on the *PROMPT PAY TYPE MAINTENANCE SCREEN (PM077)*.

8.4 Commodity Code

The required **Commodity Code(s)** are currently established in CFS on the *COMMODITY CODES MAINTENANCE SCREEN (FM002)*.

8.5 Invoice Type

The default CFS **Invoice Type** of 'CVINV' must be entered on the *ACCOUNTING SYSTEM CODE MAINTENANCE SCREEN (GL021)* under the accounting system code 'ACCR'.

8.6 Employee Information

A new employee account on the *EMPLOYEE INFORMATION MAINTENANCE SCREEN (GL029)* will need to be establish for each ASC and Headquarters. The **First Name** is the specific ASC and the **Last Name** is 'Conversion'. The system generated employee numbers will be used on the **FIMA Terminal Code** and CFS Employee crosswalk.

8.7 Document Matching

Document matching criteria must be defined on the *INVOICE MATCHING MAINTENANCE SCREEN (PM021)*. This requires that each **CFS Item Type** be matched to the corresponding **CFS Document Type(s)** as well as the default **Invoice Type** 'CVINV'.

8.8 Document Tolerance

Document tolerance levels for each **CFS Document Type** must be entered on the *PAYMENT PARAMETERS MAINTENANCE SCREEN (PM010)*.

9 Overall Risks

The following table lists risks identified during the Convert Non-CFS Undelivered Orders and Accruals Program Module analysis.

Table 12. Overall Risks

Ref	Category	Description	Mitigating Action
Risk1	CAMS Teams	A strategy must be developed to convert FIMA PCS Travel Documents.	

10 Overall Issues

- To date, no discussion occurred to address the **FUNDS BALANCE** table in CFS.
  Issue Owner(s): CAMS Design and Conversion Team, Conversion Action Team

- A crosswalk table must be created to match the **FIMA Document Type** with the corresponding CFS Document Type.
  Issue Owner(s): CAMS Design and Conversion Team

- A crosswalk table must be created to match the **Major Object Class** and **Minor Object Class** combination with the corresponding CFS **Item Type**.
  Issue Owner(s): CAMS Design and Conversion Team

- A crosswalk table must be created to match the **FIMA Document Type** with the corresponding CFS **Prompt Pay Type** and **CFS Commodity Code.**
  Issue Owner(s): CAMS Design and Conversion Team

- A crosswalk table must be created to match the FIMA **Terminal Code** with the corresponding CFS **Payment Office Code** and CFS **Employee Number**. Any FIMA Document Numbers with more than one distinct **Terminal Code** will not be converted based on the CFS system restraint of only one employee number on a transaction.
  Issue Owner(s): CAMS Design and Conversion Team

- Any FIMA Document Numbers with more than one distinct **Terminal Code** will not be converted based on the CFS system restraint of only one employee number on a transaction.
  Issue Owner(s): CAMS Design and Conversion Team

- Employee information needs to be validated and crosswalked in the appropriate Functions.
  Issue Owner(s): CAMS Design and Conversion Team

- The CFS Maximum Sequence Numbers (Maxseqnos) table must be updated each time the table is referenced for a number.
  Issue Owner(s): CAMS Design and Conversion Team

- The 30-SEP-2002 date used to populate several Purchase Order and Estimated Accrual fields are validated against the **YEAR END STATUS** and **ACCOUNTING PERIODS** table during the Validate Document Data Function.
  Issue Owner(s): CAMS Design and Conversion Team

- The **Exchange Rate** field is populated by the appropriate table in CFS.
  Issue Owner(s): CAMS Design and Conversion Team

- The **Schedule Type** field is populated by the appropriate table in CFS and validated correctly.
  Issue Owner(s): CAMS Design and Conversion Team

- The **Bank Code** field on **AP DETAIL** is validated by the appropriate table in CFS.
  Issue Owner(s): CAMS Design and Conversion Team

- The Program Module verifies that any FIMA Document Number which is identified as a No-Match document, does not have a value greater than zero in the Undelivered Orders field. A document that does not meet this criteria is not converted and appears on the error report.
  Issue Owner(s): CAMS Design and Conversion Team

- Estimated Accrual functionality will be modified for the ASAP Interface to require a vendor *payment* address for No-Match transactions. This modification affects the FIMA to CFS automated accounts payable open document conversion of No-Match documents. The program module designs will be modified to incorporate this change to the Estimated Accrual functionality.
  Issue Owner(s): CAMS Design and Conversion Team

- A strategy needs to be developed to determine the appropriate Transaction Number for Purchase Order and Estimated Accrual documents. The Transaction Code will dictate which accounts are invoked in the CFS General Ledger. An appropriate Transaction Description must also be developed.
  Issue Owner(s): CAMS Design and Conversion Team

- Any modifications to the Purchase Order tables for the Grants Implementation will have an affect on the automated accounts payable open document conversion.
  Issue Owner(s): CAMS Design and Conversion Team

- Travel advances for travel undelivered orders (Document Types 20 and 27) should be addressed in detailed design.
  Issue Owner(s): CAMS Design and Conversion Team

- What FIMA Document Type is associated with imprest funds? This issue will have to be resolved as part of detailed design.
  Issue Owner(s): CAMS Design and Conversion Team

- Transactions entered through GL005 must be appropriately marked as "Conversion" transactions so the auditors will understand.
  Issue Owner(s): CAMS Design and Conversion Team

- The CD&C Team needs to address how the program will handle situations where the data needs to be "backed out" if one of the validations in the program fails. Will the data be rolled back to a certain point, or backed out after a validation fails and then move on to the next record?
  Issue Owner(s): CAMS Design and Conversion Team

11 Assumptions

- This Program Module only provides detailed analysis for converting FIMA Document Types '01' through '49'. All other FIMA Document Types will be addressed in separate conversion efforts.

- Closed Documents on FIMA ACTDOC will be not be converted.

- The General Ledger End Date for converted documents will be defaulted to '30-SEP-2002'.

- The value of the Secondary Number in FIMA ACTDOC will not be preserved on converted documents.

- The value of the Document Sequence number will not be preserved on converted documents. An internally generated counter will assign an Item Number to replace the Document Sequence number. The counter is initialized for each FIMA Document Number and incremented as needed for each Document Sequence number.

- Data clean up will have began prior to the automated accounts payable open document conversion. Data clean up will address any document balances for invalid FIMA Document Types. This issue was agreed upon during the Conversion Action Team Meeting.

12 References

The information discussed in this document was compiled by conducting a series of user interviews, reviewing CFS functionality and documentation, and reviewing technical documentation.

For all references, please refer to the *Overview of the Automated Accounts Payable Open Document Conversion Detailed Design.*

Appendix A

The table below identifies how FIMA Document Types will be converted to CFS. Based on FIMA accounts payable document processing, an accrual, an undelivered order, or both may be required in CFS. The FIMA Document Type column lists every accounts payable Document Type present in the FIMA DTYPE table. FIMA Document Types 50 - 59 are excluded from the table since these document types require a manual conversion effort. The CFS Document Type column lists the CFS Document Type that corresponds to each FIMA Document Type. This value will appear in the Document Type field on the Purchase Order Transaction Screen when the document is converted. The Description column is the document title from the FIMA DTYPE table corresponding to each FIMA Document Type. Finally, the CFS Matching column indicates whether the FIMA Document Type will be converted as a 2-Way match or No-Match document. A 2-Way match document requires a Purchase Order and may or may not require an Estimated Accrual. A No-Match document requires only an Estimated Accrual. The Drop/Not Used designation indicates the corresponding FIMA Document Type is no longer in use and subsequently will not be converted.

Table 13. FIMA Document Matching Matrix

FIMA Document Type	CFS Document Type	Description	CFS Matching
01	CONV01	Purchase Orders	2-Way
02	CONV02	Blanket Purchase Order	2-Way
03	CONV03	Contracts - Non-recurring	2-Way
04	CONV04	Grants	Separate conversion effort
05	CONV05	Recurring Contracts - Not more than 1 FY	2-Way
06	CONV06	Contracts for Continuing Services	No-Match

Table 13. FIMA Document Matching Matrix			
FIMA Document Type	CFS Document Type	Description	CFS Matching
07	CONV07	Job Orders (GSA 1354)	2-Way
08	CONV08	Requests for Training (NOAA 53 - 1)	2-Way
09	CONV09	Printing Requisitions (CD-10)	2-Way
12	CONV12	Purchase Order/Invoice/Voucher (SF-44)	No-Match
13	CONV13	Fedstrips/Milstrips	2-Way
14	CONV14	Bankcard Purchases	2-Way.
19	CONV19	Misc. Certified Invoices	No-Match
20	CONV20	Travel Orders	Currently no strategy exists for converting travel advances.
21	CONV21	Travel Roll Voucher	Drop/Not Used
22	CONV22	Transportation Requests	Drop/Not Used
23	CONV23	Government Bills of Lading	2-Way
24	CONV24	Trip Authorizations (NOAA 42-5)	Drop/Not Used
25	CONV25	Ocean Freight Bills	Drop/Not Used
26	CONV26	Travel Order -PCS	2-Way

Table 13. FIMA Document Matching Matrix			
FIMA Document Type	CFS Document Type	Description	CFS Matching
27	CONV27	Foreign Travel Order	Currently no strategy exists for converting travel advances.
28	CONV28	Trip Authorizations for Foreign Travel	Drop/Not Used
29	CONV29	Invoices for Other Travel & Transport	Motorpool - Will determine by working with the Design/Conversion team (consider the role of the new interface).
30	CONV30	Reimb. Vouchers - Agent Cashier	No-Match
31	CONV31	Reimb. Vouchers - Other	No-Match
32	CONV32	Foreign Posts of Duty - (SF-1069)	Drop/Not Used
33	CONV33	Vouchers Related to Contract Observers	No-Match
34	CONV34	GSA Retail Store Purchases	Drop/Not Used

Table 13. FIMA Document Matching Matrix

FIMA Document Type	CFS Document Type	Description	CFS Matching
35	CONV35	Foreign Trainee Vouchers (SF-1034)	2-Way
36	CONV36	Service Station Delivery Tickets/Invoice	No Match
37	CONV37	Vouchers for Uniform Allowance	Drop/Not Used
38	CONV38	Vouchers for Commuted Subsistence	No Match
40	CONV40	Purchase Order - One Time Vendor Pymt	2-Way
41	CONV41	Purchase Order - Advance Pymt	2-Way
42	CONV42	Purchase Order - CyclicPymts	2-Way
43	CONV43	Purchase Order - Recurring Payments	2-Way
46	CONV46	Retired C. O. & Annuitants Payroll - 1329M	Drop/Not Used
47	CONV47	Family Separation Allowance Claim Auth	Drop/Not Used
48	CONV48	Claims	Drop/Not Used
49	CONV49	Other Misc. Vouchers & Claims	2-Way

Appendix B

The purpose of the following table is to document the relationship between the FIMA Object Classes and CFS Item Types for each FIMA Document Type. With the exception of Object Class 21-48 for FIMA Document Type 26, all other combinations were present in FIMA ACTDOC as of October 30, 2001. Major and Minor Object Class combinations resulting in identical CFS Item Types are grouped together. For example in FIMA Document Type '06', the Object Class combinations 23-31, 23-37, 23-38, and 23-39 are attributed to the same CFS Item Type, 'UTIL'. For this reason they are displayed in the table as 23- 31, 37, 38, 39.

CFS Item Types must be established on GL021, the Accounting System Codes Maintenance Screen, prior to conversion. In addition the CFS Conversion Document Type and CFS Item Type relationship must be defined on PM021, the Invoice Matching Maintenance Screen.

KEY:

ANY - This encompasses all Object Classes which could fall under the column heading.

ALL OTHER - This represents the default CFS Item Type that will be used if the criteria within the given FIMA Document Type are not met.

BUILD* - Items in ***bold italics*** denote an Item Type not currently established in CFS.

Table 14. FIMA Item Type Matrix

FIMA Document Type	Proposed CFS Conversion Document Type	Major Object Class	Minor Object Class	CFS Item Type
01	CONV01	ANY	ANY	INTAGR
02	CONV02	22	ANY	EXMAIL

Table 14. FIMA Item Type Matrix				
FIMA Document Type	Proposed CFS Conversion Document Type	Major Object Class	Minor Object Class	CFS Item Type
		25	ANY	SERV
		26	ANY	SUPPLY
		ALL OTHER	ALL OTHER	SERV
03	CONV03	22	ANY	FR
		23	20	LEASE
		25	ANY	SERV
		26	ANY	GOODS
		31	ANY	EQUIP
		32	10	***LAND****
		32	20	***BUILD****
		ALL OTHER	ALL OTHER	SERV
05	CONV05	25	28	STORAG
		ALL OTHER	ALL OTHER	SERV
06	CONV06	23	20	LEASE
		23	30	SERV
		23	31,37,38,39	UTIL
		25	ANY	SERV
		26	ANY	GOODS
		31	23	EQUIP
		32	30	LEASE

Table 14. FIMA Item Type Matrix

FIMA Document Type	Proposed CFS Conversion Document Type	Major Object Class	Minor Object Class	CFS Item Type
		32	31	LEASE
		ALL OTHER	ALL OTHER	SERV
07	CONV07	ANY	ANY	RWA
08	CONV08	ANY	ANY	TRAIN
09	CONV09	ANY	ANY	***PRINT****
12	CONV12	26	ANY	GOODS
		ALL OTHER	ALL OTHER	SF44
13	CONV13	26	13, 15, 17	MILSTR
		ALL OTHER	ALL OTHER	FEDSTR
14	CONV14	ANY	ANY	PCARD
19	CONV19	ANY	ANY	MISC
20	CONV20	21	43	GTADOM
		ALL OTHER	ALL OTHER	DOM
23	CONV23	ANY	ANY	GBL
26	CONV26	21	43, 48	GTAPCS
		ALL OTHER	ALL OTHER	PCS
27	CONV27	21	48	GTAFOR
		ALL OTHER	ALL OTHER	FOR
29	CONV29	ANY	ANY	***MTRPL****
30	CONV30	11	59	CD326

Table 14. FIMA Item Type Matrix

FIMA Document Type	Proposed CFS Conversion Document Type	Major Object Class	Minor Object Class	CFS Item Type
		ALL OTHER	ALL OTHER	NF34-6
31	CONV31	ANY	ANY	SF1164
33	CONV33	ANY	ANY	OBSERV
35	CONV35	ANY	ANY	STIPEN
36	CONV36	ANY	ANY	GASCD
38	CONV38	ANY	ANY	SERV
40	CONV40	22	ANY	FR
		23	20	LEASE
		23	30	SERV
		24	ANY	***PRINT****
		25	ANY	SERV
		26	ANY	GOODS
		31	ANY	EQUIP
		32	10	***LAND****
		32	20	***BUILD****
		ALL OTHER	ALL OTHER	SERV
41	CONV41	25	ANY	SERV
		26	ANY	GOODS
		ALL OTHER	ALL OTHER	SERV
42	CONV42	25	ANY	SERV

Table 14. FIMA Item Type Matrix				
FIMA Document Type	Proposed CFS Conversion Document Type	Major Object Class	Minor Object Class	CFS Item Type
		26	ANY	GOODS
		ALL OTHER	ALL OTHER	SERV
43	CONV43	22	ANY	FR
		23	20	LEASE
		23	30	SERV
		24	ANY	***PRINT****
		25	ANY	SERV
		26	ANY	GOODS
		31	ANY	EQUIP
		ALL OTHER	ALL OTHER	SERV
48	CONV48	ANY	ANY	SF1034
49	CONV49	ANY	ANY	MISC

Appendix C

The following table contains all the active FIMA Document Types. For each FIMA Document Type, the default Prompt Pay Type(s) for that Document Type is indicated. Based on the Prompt Pay Type, the default Commodity Code(s) is listed. The Convert Non-CFS Undelivered Orders and Accruals Program Module will use this matrix to determine the Commodity Code (and Prompt Pay Type) for each Document Type.

Document Type '03' lists two possible Prompt Pay Types. For this Document Type, a distinction must be made as to which Prompt Pay Type is assigned. Under the following conditions, Document Type '03' will be considered a CWIP contract and will obtain a Prompt Pay Type value of 'CONSTR' and a corresponding Commodity Code 'CONSTR':

- All documents of Document Type '03' in the '4' fund.
- All documents of Document Type '03' in the 'Q' fund with an 'F' in the third position of the FIMA Task Code.
- All documents of Document Type '03' in the '2' fund with an 'F' in the third position of the FIMA Task Code.
- All documents of Document Type '03' in the 'V' fund with an 'F' in the third position of the FIMA Task Code.

If the document does not satisfy one of the above conditions it will not be considered a CWIP contract and will obtain a Prompt Pay Type value of 'STD' and a corresponding Commodity Code 'G/S'.

CFS PROMPT PAYMENT TYPES AS OF NOVEMBER 12, 2001

ADVPAY	IMMEDIATE PAYMENT	NONE	Y	15-APR-1999
AGR	PERISHABLE AGRICULTURE COMMODITIES	AGR	Y	09-SEP-1996
CONSTR	CONSTRUCTION CONTRACT (PROGRESS PAYMENTS)	CONSTR	Y	15-APR-1999
DAIRY	DAIRY PRODUCTS, EDIBLE FATS, OILS	DAIRY	Y	09-SEP-1996
G/S	GOODS OR SERVICES	STD	Y	20-FEB-1997
MEAT	MEAT OR MEAT FOOD PRODUCTS, POULTRY, EGG	MEAT	Y	09-SEP-1996
MISC	MISCELLANEOUS G/S NOT SUBJECT TO PPA	NONE	Y	21-APR-1999
RAT	RATIFICATION	NONE	Y	15-APR-1999

Table 15. FIMA Commodity Code Matrix			
FIMA Document Type	Document Type Description	Prompt Pay Type	Commodity Code
01	Purchase Orders	NONE	MISC
02	Blanket Purchase Order	STD	G/S
03	Contracts - Non-recurring	STD	G/S
03	Contracts - CWIP	CONSTR	CONSTR
05	Recurring Contracts - Not more than 1 FY	STD	G/S
06	Contracts for Continuing Services	NONE	MISC
07	Job Orders (GSA 1354)	NONE	MISC
08	Requests for Training (NOAA 53 - 1)	STD	G/S
09	Printing Requisitions (CD-10)	NONE	MISC
12	Purchase Order/Invoice/Voucher (SF-44)	STD	G/S
13	Fedstrips/Milstrips	NONE	MISC
14	Bankcard Purchases	STD	G/S
19	Misc. Certified Invoices	STD	G/S
20	Travel Orders	NONE	MISC
23	Government Bills of Lading	STD	G/S
26	Travel Order -PCS	NONE	MISC
27	Foreign Travel Order	NONE	MISC
29	Invoices for Other Travel & Transport	NONE	MISC

Table 15. FIMA Commodity Code Matrix

FIMA Document Type	Document Type Description	Prompt Pay Type	Commodity Code
30	Reimb. Vouchers - Agent Cashier	NONE	MISC
31	Reimb. Vouchers - Other	NONE	MISC
33	Vouchers Related to Contract Observers	NONE	MISC
35	Foreign Trainee Vouchers (SF-1034)	NONE	MISC
36	Service Station Delivery Tickets/Invoice	STD	G/S
38	Vouchers for Commuted Subsistence	NONE	MISC
40	Purchase Order - One Time Vendor Pymt	STD	G/S
41	Purchase Order - Advance Pymt	NONE	ADVPAY
42	Purchase Order - CyclicPymts	STD	G/S
43	Purchase Order - Recurring Payments	STD	G/S
49	Other Misc. Vouchers & Claims	STD	G/S

Appendix D

The following table contains all the active FIMA Terminal Codes. For each FIMA Terminal Code, the corresponding CFS Payment Office Code is displayed. The Convert Non-CFS Undelivered Orders and Accruals Program Module will use this matrix to determine the Payment Office Code for each Terminal Code. The Payment Office Code will subsequently drive the employee information required for several fields within CFS.

Table 16. FIMA Payment Office Code Matrix

FIMA Terminal Code	CFS Payment Office Code
1	HQS
2	HQS
3	WASC
4	MASC
5	EASC
6	WASC
7	HQS
8	HQS
9	CASC
10	HQS

Appendix E

Document Modifications

Vendor Processing

Due to new information regarding how CFS handles payments and payment addresses, this module must change how it validates and uses the vendor(s) associated with each document converted. If the document is a Two-Way Match document, CFS requires that both a purchase and payment address be added for the vendor. To handle this requirement, the modules validates that both the **Purchase Vendor ID** and the **Payment Vendor ID** on **CONV_ACTDOC** is present and active in CFS. If the document is a No-Match document, CFS only requires a payment address to be added for the vendor, meaning that the module only validates that the **Payment Vendor ID** on **CONV_ACTDOC** is present and active in CFS.

The Record Purchase Order Function populates CFS obligation documents with the **Purchase Vendor ID.** In addition the Function uses the **Purchase Vendor ID** to select all vendor data from the CFS vendor tables.

The Record Estimated Accrual Function populates an Estimated Accrual with the **Payment Vendor ID.** In addition the Function uses the Payment Vendor ID to select all vendor data for the CFS vendor tables.

General Ledger Posting

The Program Module follows the CFS general ledger posting logic at the time of conversion for CFS Obligation documents and Estimated Accruals. Following the CFS general ledger posting logic requires the Program Module to use the Transaction Codes established on the *General Ledger Transaction Code Maintenance Screen (GL022)*.